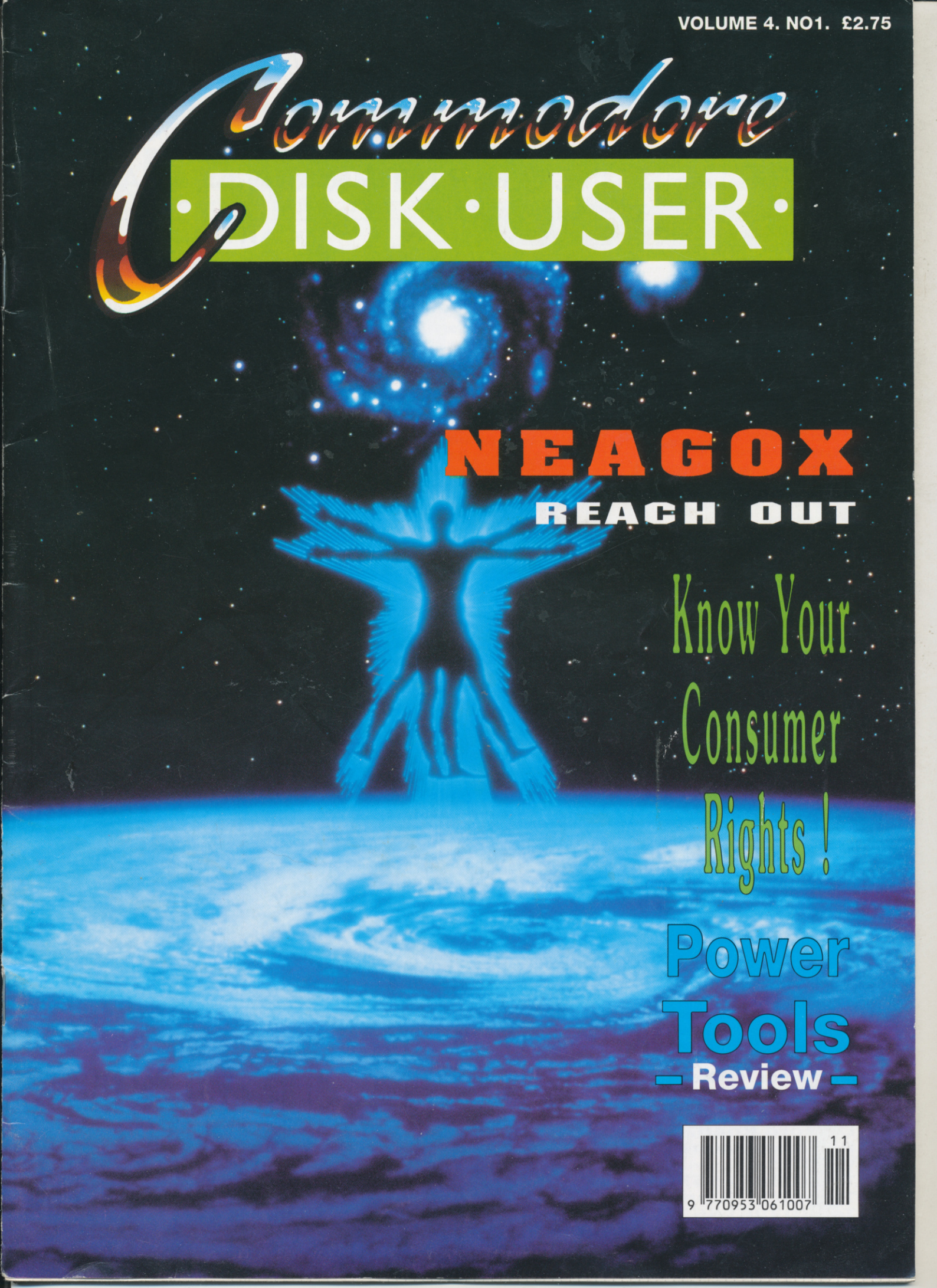
VOLUME 4. NO1. £2.75

Commodore DISK USER



NEAGOX

REACH OUT

Know Your Consumer Rights !

Power Tools

– Review –

9 770953 061007 11

f/s/s/l
COMPUTER SOFTWARE

SUPER SNAPSHOT v5

Well, it may have taken a few years of hard work and five powerful versions, each one breaking new ground, but Super Snapshot has become the best cartridge in the world. The list below details the main features of Super Snapshot v5; if you need a little more persuasion look back to CDU issue 19, you'll be impressed.
So take a look at the red box you've got plugged in, and if our specifications knock it for six or you don't own a cartridge then don't just sit there, buy Super Snapshot v5 today!

FEATURES:

- ► All features available at the press of a button
- ► Works with all 64 (c) and 128 (D) computers
- ► Compatible with 1700/1764/1750 REU'S
- ► Snap any memory resident program into 1 file
- ► Save 7× faster and load 15× faster on the 1541, 1571 & 1581. Speeds of up to 25× faster when using Turbo 25 – even faster than Replay
- ► Super DOS Wedge
- ► GAME MASTER menu with sprite killer, infinitive lives generator and joystick port swapper
- ► Programmable function Keys
- ► Sprite Monitor
- ► Exclusive Character Set Monitor
- ► Exclusive Sound Sample Monitor
- ► Exclusive Boot sector support
- ► 300/1200/2400 Terminal program (40/80 column)
- ► SUPER DISK SNAPSHOT – our new super nibbler
- ► SCREEN-COPY now loads or saves in more formats and dumps in COLOUR to STAR LC10C printers and in 16 grey scales
- ► Improved full featured M/L monitor that DOES NOT CORRUPT MEMORY. Interrupt, examine and resume any running program
- ► Drive Mon
- ► BASIC PLUS with 15 new BASIC commands
- ► FILE MANAGEMENT SYSTEM – scratch, unscratch, rename or adjust skew. Includes our 1 or 2 drive file copier with partition support for the 1581
- ► Fast disk copier, 1 or 2 drives
- ► OUR FILE COPIER, DISK COPIERS and NIBBLER MAKE FULL USE OF THE REU'S
- ► Sequential file reader
- ► Utility disk
- ► Plus 150+ Kracker Jax parameters

ONLY £34.95

VISA Access Prices include VAT & U.K. delivery. Overseas orders send advertised price plus £2.50 for Airmail. Please send cheque, Postal Order or Credit Card details. VISA/ACCESS orders accepted by 'phone.

Masons Ryde,
Defford Road,
Pershore, Worcs.
WR10 1AZ.
Tel: (0386) 553153
Technical Support
Tel: (0386) 553222

ANIMATION STATION

Unleash your creativity with Animation Station, a powerful utility for generating all kinds of graphics on your Commodore 64 or 128. Built-in, predrawn pictures give you a head start on your creations. Automatic generation of circles, ovals, squares, boxes, straight lines, typography and other geometric shapes gives you all the tools of a self-contained, electronic drafting room. Combine type and graphics on the screen, draw in many colours, even connect your VCR to create titles and graphics for your home movies. Screen dump to your printer. Koala Compatible. Package includes design touch pad, pencil and graphics software. Ideal for GEOS users.

ONLY £59.95

VIDEO BYTE 3

Digitize video images from your VCR, laser disk, B/W or colour camera, off the air or cable TV. New version 3.0 software features full re-display with multi-capture mode, menu select printing, expanded colourizing features, save to disk and much more. The hardware is no larger than an average cartridge which plugs into the User Port. The menu driven capture software is easy to use and pictures stored to disk can be imported into most popular drawing packages including GEOS. Prints in full colour to the Star LC10 when used with SuperSnapshot.

ONLY £79.95

HOME VIDEO PRODUCER 64

Nothing can make home videos so special. Add titles, text, and even brilliant graphics to your favourite home videos with ease and the help of the Home Video Producer. You have the choice of 10 typefaces, 75 large full-colour graphics and ready made segments, but the most appealing aspect of the Home Video Producer is the ease with which you can do all this.

ONLY £29.95

C64 SLIMLINE UPGRADE CASE

Make your older C64 look like a newer model! All you need is a screwdriver and about 15 minutes to transfer your C64 insides to this new case. Complete instructions included.

ONLY £12.95

CARTRIDGE PORT EXTENDER CABLE

Are you cramped for space behind your computer? Is it hard to reach your cartridge port to plug-in or swap cartridges? This handy cable is the solution. The Cartridge Port Extender Cable connects to the cartridge port in the back of the computer and lets you plug in your cartridge to its other end. Since the cable is flexible, you can locate the cartridge up to 11" away for easier access. Not for REU's.

ONLY £19.95

GEOS APPLICATIONS

GEOSCAN ART

This special type of GEOS art has been created using The HandyScanner 64. Pictures are scanned at 400 dpi from magazines, books and papers to create the first geoSCAN ART Collection entitled The British Countryside and is packed full with Eagles, Owls and Butterflies.

ONLY £6.95

GEODIRECTORY

A comprehensive book listing all available GEOS programs. Details for each program is given including version numbers. The geoDirectory is divided into sections covering Paint, Write, Spell, File, Calc, Chart, Terminal, Graphic, Music, Animation, Games and many more.

ONLY £6.95

GEOTRONIX

A professional PCB designer utilizing the GEOS environment. Five double sided disks supply geoPublish with pre-designed components, sockets, edge connectors and layout grids in Photo Scraps. Using the Photo Manager and geoPublish the circuit is designed and printed.

ONLY £39.95

GEOS 64 V2.0	£29.95
GEOCALC 64	£24.95
GEOFILE 64	£24.95
GEOPROGRAMMER	£29.95
DeskPack Plus	£19.95
GEOCHART	£19.95
GEOS 128 V2.0	£39.95
GEOCALC 128	£29.95
GEOFILE 128	£29.95
GEOPUBLISH	£29.95
FontPack Plus	£19.95
Int. Font Pack	£19.95

Commodore DISK USER

CONTENTS

Volume 4 Number 1 November 1990

IN THE MAGAZINE

Welcome 4
Editors comment and instructions

Computer Time 7
The History of the Computer is told

Adventure Writing 9
We progress onto Solutions and Objects

Contributions 14
How to send in submissions to CDU

Power Tools 15
We review the latest utility from Diamond Bytes

Serving the Public 21
We highlight the success of a local store

Worth the Hassle? 34
Your rights are put under the microscope

Techno Info 38
More letters from the mailbag

65xx Interfacing 42
This fine series comes to a close

ON THE DISK

War at Sea 5
C128 version of Battleships

Full Disk Jacket 6
The third in the 'ED series

Neagox 8
Blast everything that moves

Numdef 14
A Basic game to test your reflexes

Memory Scanner 19
Look through memory the easy way

Money 64 20
Budget planning for the 90's

Xinout 23
An alternative input routine

Calendar C128 31
No more buying calendars with this utility

Gomoku 32
A nice variation of GO

Smooth Scroll Demo 37
Create your own pleasing scrolls

Publisher: Hasnain Walji
Group Editor: Paul Eves
Technical Assistant: Jason Finch
Publishing Consultant: Paul Crowder
Consultant Editor: Stuart Cooke
Advertisement Manager: Cass Gilbert
Designer: Mark Newton
Distribution: Seymour Press Distribution Ltd. Winsor House, 1270 London Road, Norbury, London SW16 4DH. Tel:081 679 1899. Fax: 081 679 8907
Printed By: Gibbons Barford Print

Subscription Rates

UK	£33.00
Europe	£39.00
Middle East	£39.30
Far East	£41.60
Rest of World	£39.70 or $69.00

Airmail rates on request
Contact: Select Subscriptions. Tel: (0442) 876661

Commodore Disk User is a monthly magazine published on the 3rd Friday of every month. Alphavite Publications Limited, 20, Potters Lane, Kiln Farm, Milton Keynes, MK11 3HF. Telephone: (0908) 569819 FAX: (0908) 260229. For advertising ring (0908) 569819

Opinions expressed in reviews are the opinions of the reviewers and not necessarily those of the magazine. While every effort is made to thoroughly check programs published we cannot be held responsible for any errors that do occur.

The contents of this publication including all articles, designs, drawings and programs and all copyright and other intellectual property rights therein belong to Alphavite Publications Limited. All rights conferred by the law of copyright and other intellectual property rights and by virtue of international copyright conventions are specifically reserved to Alphavite Publications Limited and any reproduction requires the prior written consent of the company

© 1990 ISSN 0953-0614

INSTRUCTIONS

EDITORS COMMENT

You will find in this issue quite a few features to while away the cold evenings. We have rather an interesting article on the "Rights" of the purchaser, and the attitude of some companies towards those "Rights". There is an article on the success of a local computer store in Hemel Hempstead. STEVE CARRIE finishes his series on INTERFACING with the 65xx, whilst YASIN POPTANI provides us with a neat little, build your own, joystick autofire routine. C128 users once again have a couple of programs to keep them busy. WAR AT SEA is a nice variation on the Battleships theme. Good news for people that have been wondering where the series on "MULTITASKING ON THE C128" has got to. Due to earlier technical problems, we had to postpone this series. I am happy to say that as from the DECEMBER issue, we will be starting it again. Apologies to DAVID KELSEY for this somewhat extended delay. Apologies also go to all the followers of our `C' series. Unfortunately, there is no offering in this months issue. I hope that there is something in this months issue to please everyone. Without a further of a do, let's get on with business.....

DISK INSTRUCTIONS

Although we do everything possible to ensure that CDU is compatible with all C64 and C128 computers, one point we must make clear is this. The use of 'Fast Loaders', 'Cartridges' or alternative operating systems such as 'Dolphin DOS', may not guarantee that your disk will function properly. If you experience problems and you have one of the above, then we suggest you disable them and use the computer under normal, standard conditions. Getting the programs up and running should not present you with any difficulties, simply put your disk in the drive and enter the command.

LOAD"MENU",8,1

Once the disk menu has loaded you will be able to start any of the programs simply be selecting the desired one from the list. It is possible for some programs to alter the computers memory so that you will not be able to LOAD programs from the menu correctly until you reset the machine. We therefore suggest that you turn your computer off and then on again, before loading each program.

HOW TO COPY CDU FILES

You are welcome to make as many of your own copies of CDU programs as you want, as long as you do not pass them on to other people, or worse, sell them for profit. For people who want to make legitimate copies, we have provided a very simple machine code file copier. To use it, simply select the item FILE COPIER from the main menu. Instructions are presented on screen.

DISK FAILURE

If for any reason the disk with your copy of CDU will not work on your system then please carefully re-read the operating instructions in the magazine. If you still experience problems then:

1. If you are a subscriber, return it to:
Select Subscriptions Ltd
5, River Park Estate
Berkhamsted
Herts
HP4 1HL
Telephone; 0442 876661

2. If you bought it from a newsagents, then return it to:
CDU Replacements
Interceptor Group
Mercury House
Calleva Park
Aldermaston
Berks
RG7 4QW
Telephone; 0734 817421

Within eight weeks of publication date disks are replaced free.

After eight weeks a replacement disk can be supplied from INTERCEPTOR GROUP for a service charge of £1.00. Return the faulty disk with a cheque or postal order made out to INTERCEPTOR GROUP and clearly state the issue of CDU that you require. No documentation will be supplied. Please use appropriate packaging, cardboard stiffener at least, when returning disk. Do not send back your magazine, only the disk please.

NOTE: Do not send your disks back to the above address if its a program that does not appear to work. Only if the DISK is faulty. Program faults should be sent to:
BUG FINDERS, CDU, Alphavite Publications Ltd, Unit 20, Potters Lane, Kiln Farm, Milton Keynes, MK11 3HF. Thank you.

WAR AT SEA!

Got yourself a C128? 80 and 40 Column monitors? Disk drive and two joysticks? then play WAR AT SEA!

PAUL TRAVNOR

WAR AT SEA! is a two player game for the C128 loosely based on the old favourite BATTLESHIPS. It uses the unique feature of the C128's two screen outputs. You will need separate 80 column and 40 column monitors (the 40 column monitor can be substituted by a ordinary T.V.). These will need to be placed so that each player can not see his opponents screen, player one will use the 80 column screen along with joystick 1 and player 2 the 40 column screen with joystick 2.

SETTING UP

When you start the game each player must position their fleet, player one goes first, their 80 column screen will display the instruction 'Position your Fleet' along with a list of the vessels. Each fleet consists of five vessels, each type of vessel has a corresponding colour and are also covers a different number of grid squares. The chosen colours should be readable on a monochrome monitor, being able to distinguish between them is not critical to playing the game.

TYPE	COL.	LEN	No of.
Aircraft Carrier	Light Green	4sq	1
Destroyer	Cyan	3sq	1
Submarine	Yellow	2sq	1
Frigate	Light Blue	1sq	2

Positioning is done on the main grid, starting with the largest vessel, by moving the joystick controlled cursor around the grid. You press the fire button to mark one end of the vessel and then you just need to move the joystick in one of four directions (up, down, left or right) to position the rest of the vessel. All vessels are positioned in this way except for the two frigates which only take up one grid square and therefore you only have to move to the desired position and press fire. After player one has positioned their vessels the instruction for player 2 to do theirs will appear on their screen. If when setting up you move off the grid with the joystick controlled

cursor you will automatically appear on the opposite side of the grid. It is also impossible to set up a vessel that either goes off the grid or crosses a existing vessel position, if you try to lay one vessel across another it will just be cancelled and you will have to start positioning that vessel again.
Once both players are ready the details of vessel positions will be transferred from the main grid to a smaller grid in the bottom right hand corner of each screen. This smaller grid will then be used to keep track, automatically, of your opponents attempts to destroy your fleet. The main grid is then free for you to carry out your attacks.

GAMEPLAY

Player one goes first, using the joystick you can move around the grid pressing the fire button when over the square you wish to attack. You will then hear your missile launch into the air and if you hit a vessel you will hear an explosion as the missile makes contact. The grid display will change from its original dot to a small cross for a miss, or a large coloured cross for a hit (the colour represents the type of vessel). The mini grid on your opponents display will also show your attacks as they happen. As with setting up if you move the cursor off the grid you will automatically appear on the opposite side. Once a grid square has been attacked it is then impossible to attack again, the fire button will be ignored when over such squares. Each player gets one attempt at a time and your screen will show in the right hand top corner the message 'Aim and Fire' when it is your turn. The complete destruction of a vessel will be signalled by a larger explosion along with name of the vessel appearing in your list of 'Enemy Vessels Destroyed' on the right hand side of your screen. Also on the right hand side is a complete numeric record of shots, hits and misses. The game is ended when one of the fleets is completely destroyed, you will then be given the option to start again or quit.

FULL DISK JACKET

If you want to make your own disk sleeves that includes all the filles on your disk, then this utility is for you

MIKE GREGORY

FULL DISK JACKET was written on a rainy day. I was searching through some of the older disks in my collection trying to find a half-remembered utility. I was becoming more and more frustrated at not knowing what was on each disk and also annoyed by the way the disk jackets had been warped by the labels. I'm sure you've all been there! A friend had shown me a template for a disk jacket which was prepared within a word-processor. While the concept had a few disadvantages from my own point of view, it was becoming more attractive. I finally resolved to do something.

FULL DISK JACKET is the end result. It makes use of various printer features to produce a disk jacket template which has the directory printed in three columns. For those of us who use 'flippies', the directory of the back of the disk is also printed. When the template is cut out and folded, this second directory ends up printed the right way instead of upside down! The completed jacket is intended to hold the disk and its manufactured jacket.

```
  FULL DISK JACKET
 COPYRIGHT MIKE GREGORY, 1990

  MAKE SURE DRIVE AND PRINTER ARE BOTH
SWITCHED ON

  INSERT DISK AND PRESS ANY KEY WHEN
READY

***READING NEW CHARACTERS

***READING DISK DIRECTORY

  DO YOU WANT TO DELETE ANY FILENAMES
  (Y/N)?
```

The programme is written in Basic so that it is easy to change to suit a user's own equipment. The programme will run without amendment on a Gemini-10X printer with a Cardco /? interface.

```
  DO YOU WANT TO DELETE ANY FILENAMES
  (Y/N)?

5     MEM SCAN              PRG    D OR CR?
1     SPR.                  PRG    D OR CR?
13    MEMORY SCANNER        PRG    D OR CR?
14    XINOUT                PRG    D OR CR?
25    XINOUT.CODE           PRG    D OR CR?
109   NEAGOX                PRG    D OR CR?
24    NUMBER DEFENDER       PRG    D OR CR?
35    FULL DISK JACKET      PRG    D OR CR?
4     SETUP CALENDAR        PRG    D OR CR?
4     CAL.JAN               SEQ    D OR CR?
4     CAL.FEB               SEQ    D OR CR?
4     CAL.MAR               SEQ    D OR CR?
4     CAL.APR               SEQ    D OR CR?
4     CAL.MAY               SEQ    D OR CR?
```

Any other interface should work provided that it will allow control characters to be sent unchanged to the printer. This is normally arranged via a 'transparent' mode in the interface. As indicated, in this mode, Ascii commands are not changed by the interface and are sent directly to the printer.

Other printers might be usable if the following print features are available:

a) Downloadable characters...essential for the upside down font. The Gemini-10X uses one byte to designate the particular character, another to indicate whether or not it has a descender plus a further nine bytes to provide the character bit-map. This makes a total of eleven bytes per character. Other printers may require more bytes. Within the programme, the DATA statements are laid out on a character-by-character basis so as to allow changes to be made more easily

b)Subscript/superscript...not essential. Allows the three-column directory print-out within the template bounds. If this mode is not available, a two-column print-out could be programmed.

c)Variable line spacing...not essential. The number of printed lines would need to be changed so as to keep the template dimensions constant.

No attempt has been made to reproduce the Commodore graphic characters, mainly because there is a 128 character limit in the downloadable system. If a programme or file name contains a graphic, the print-out will show it as an asterisk.

COMPUTER TIME

Take a journey through time and learn all about how that grey box you love so much, known to you as your computer, came to be

STEVEN BURGESS

When I was alive, the history of computing was included in the 'O' level syllabus and formed a small part of the exam. Of course the information wasn't exactly essential but it was interesting and nothing else was lost because of it. These days, of course, it is not included. The syllabus is directed more to practicality. And history isn't practical, is it?

Anyway, I plan to redress this by delving, not too deeply, into the history of our mutual friend. A bit of extra knowledge doesn't hurt, does it?

IN THE BEGINNING

Every since man has had the need to count he has used a method of doing so to simplify the process. This may involve counting on his fingers or with pebbles or by scratching on rock to form tally marks or by making marks in the sand.

When man became more sophisticated his method of counting became more sophisticated. The abacus was invented. Nobody is entirely sure when but it is known to have been around about 3000 BC and has quite remarkably stood the test of time; in some eastern countries it is still used by many people with great efficiency.

However, after that point not a great deal happened with regard to making the calculation of maths - even simple maths - easier. You simply had to use your head. But unfortunately not many people were terribly well educated mathematically.

They struggled on, though, until 1614 when John Napier invented logarithms (which we all remember from school - those little orange books packed with figures with mind boggling decimal places). The theory behind logarithms was for every number you had another number called a logarithm. If the logarithms of two numbers were added together then the anti-logarithm of the result was the product of the original numbers, that is the two original numbers multiplied together. Subtracting the logarithms gave numbers divided. Only 3 years later Napier had another invention, again

"The slide rule is born"

based on the addition of numbers to bring about the multiplication. He invented Napier's bones, which were rods on which multiplication tables were printed. You chose the appropriate rods for the numbers you were to multiply and then read off the numbers to find the result.

William Oughtred, as an extension to Napier's idea, then invented the slide rule, many of which are still around now (I think we've got one upstairs tucked in a drawer, still) but not used since the calculator came about. Mechanisation came later. Blaise Pascal, in order to help his father who was a tax man, invented the Pascaline - the world's very first calculating machine. Capable of addition and subtraction, the machine used cogs and wheels to calculate sums. The figures were dialled on the wheels and the result shown at the top. Unfortunately, due to the expense of the machine and the current attitude, Pascal's machine was not comercially successful. However it did mark the first real step on our road to the modern computer.

Gottfried Leibniz entered the game after Pascal with an extension to his machine. It was capable of multiplication and division. It incorporated a stepped wheel which proved to be a very significant step forward (if you'll pardon the pun). The machine was operated by a handle which, after entering the appropriate figures on wheels, was turned forward for multiplication and in reverse for division. This too was not commercially successful, but only because of the inventor losing interest. The next big step was to come from the mathematician George Boole. Unless you've not already guessed he developed boolean algebra which was concerned more with symbols rather than numeric quantities in the solving of problems.

"The Father of Computers"

Next came the man known as the Father of Computing. Charles Babbage invented first the difference engine. The machine was invented more in theory than in practice, being so utterly complicated that any minor imperfections had enormous repercussions. Unable to get any more money from the government to finance his machine, he was unable to complete it fully to his satisfaction. However his mind did not stop there. In this period, Babbage went onto bigger and better things. He developed, in his mind, a machine which was called the Analytical Engine. The machine from which your humble 64 (and indeed all computers) is decended. Babbage had invented the computer.

Unfortunately, again, his brain child never got to the stage of completion. However, its very conception started the rocky road to the invention of the computer. The analytical engine was a machine capable of any type of calculation. It could store a program (by a means invented by Joseph-Marie Jaquard of the Jaquard loom fame - using punched cards) and output a result.

In 1890 the United States census was completed much quicker than the 1880 census. Why? Simply because a Herman Hollerith had invented a

machine, called the Hollerith Tabulator, to simplify it. It marked the first real use of punched cards in the processing of information. Interestingly Hollerith formed a company called the Tabulating Machine Company, which later became IBM. In 1824 William Thomas was born. When he was in his sixties he invented a computer to predict the tides for any given port. The output was a plotter.

From that moment on no real developments were made until 1936 when Alan Turing, mathematician, published "On Computable Numbers", considered to be the single most important work in the development of computer science. He later went on to help build the first operational computer, Colossus 1, which was used to crack codes during the war. Just before Colossus 1 though came a truly enormous computer built in Harvard University. Called the Harvard Mark I. It was built at enormous cost, which the university could not afford until helped by IBM. The computer could add two numbers in three-tenths of a second.

ENIAC (Electronic Numerical Integrator And Calculator) was the next major development. It was designed initally to calculate the trajectories of bombs - the impetus of war was very great - but eventually could be used for any computable purpose. The machine used radio valves and a tremendous amount of electricity to do its job. However, it started the computer boon.

"Lancashire leads the way with the Manchester Mk 1"

Later the idea for EDVAC came about which had the advantage of a program store, but this was never completed.

The first computer to have an editable store was the Manchester Mark I. It was invented by Prof. Tom Kilburn and Sir Frederick Calland Williams. The first computer which was commercially successful was UNIVAC, made by the same people that invented ENIAC. The UNIVAC was most innovative in that it made use of magnetic tape. It sold over 50, the first being to the US census bureau. Then came the invention of transistors which speeded up computers and made them smaller. But the most important development which furthered both of these was, of course, the silicon chip.

This lead onto computers like the sinclair ZX range and then, by incorporating more memory and more sophisticated operating systems, better computers. The home computer market exploded into life. But behind this more and more powerful computers were being developed which nobody would be able to afford to buy. Large mainframe computers occupying full rooms.

And now, in the nineties, we look back. Now, we have computers more powerful than the early ones which are stored in watches. We have mainframes more powerful than ones which once occupied rooms stored in a small cupboard. Computers are progressing all the time. Biological computers are being developed - as fast as the brain - unlimited storage capabilities. How much further can we go?

NEAGOX

Get out those joysticks and start blasting for all your worth

BRIAN SCHAU

speed, action and th simple shoot'em up in which the only way of scoring points is to shoot aliens. So the better and faster your reflexes is the more points you'll get.

THE NEAGOX PLOT

In the year 2020 the earth has become so polluted that most people are dying of different strange diseases. Every time a new child is born it's mutated.
Therefore the earth government decided that the earth population had to move to some other planet if mankind should survive. Three years later, in the year 2023, the first planet was colonated. From then on mankind colonized more than 11 planets in 30 years. Your planet NEAGOX was colonized as planet no. 8 in the year 2044. For some 40 years the on earth ... It was the Zyxonions. They came in peace and people had belief in them. But then oneday the Zyxonions showed their real intentions and took over the world
The Zyxonions took over the world government and you could no longer live a decent life because the Zyxonions took control over everything. The governments of the colonized planets joined forces and made a defensive pact which involved a group of special fighter pilots. This group was called TERRORFORCE (032) and the headquarters were situated on NEAGOX.
A couple of years after this pact was made, the Zyxonions began to attack the colonized planets. The newly trained TERRORFORCE fighter pilots were sent to destroy the Zyxonions but they were destroyed one by one. Now, this is where you take over!!!
You are the last of the fighter pilots and you have to win the final battle on your planet NEAGOX. Do you have what it takes to free the human race from enslavement of the Zyxonions????

CONTROL IN YOUR HANDS

Ok, now a few words on the controls in this game. Once the titlescreen is shown press fire to enter game mode. In the game you use joystick to steer your space craft and fire to release a missile. The joystick must be in port 2.
If you manage to get your name on the hi-score table joystick up and down flips through the alphabet and joystick left and right moves your cursor from side to side on the line. Fire stops writing.
In game mode space-bar will put you in pause mode. Fire will resume play. A space craft is earned at the end of every level.
Enjoy the game ...

The series is well under way now. We continue at looking at Objects and Solutions

ADVENTURE WRITING

JASON FINCH

First, before I go any further I would like to make an announcement! You may have noticed that neither the August issue nor the last issue, October, contained any information pertaining to this Adventure Writing series. That is because the series is going bimonthly. I shall now explain the three reasons why. Firstly, I store all my information for this sort of thing and the programs on three and a half inch disks and my 1581 has recently gone in for repair, therefore I couldn't bring you the programs that were necessary. Secondly, it gives me a bit more time to concentrate on the next article and the programs in it; and thirdly, and probably most importantly, it will give you more time to experiment with the programs and get to grips with the ideas and theory that I present. I know that a lot of readers are unable to devote as much time as they would like to CDU and therefore, by giving you two months to let everything sink in, you have no excuses! The next article after this, then, will appear in the January 1991 issue of CDU.

SOLUTION and OBJECTS

Right then, down to business! This month's installment of adventure writing tuition will begin with the topic that we left off from in September - solution and collection objects. These are essentially treated the same by the program and so I shall simply refer to them as the objects, unless there is a definite need to differentiate. One other thing to remember is that in most adventures each object has just one function or use. A player should not be expected to keep the piece of wood after it has been used to shift the boulder (example from the last article). Similarly, a key used to open one door should not be required to open another door somewhere else, and so on.

Each object has three, or possibly four, separate components. These are the detailed description, the simple noun and the object's location. An optional extra is its weight or mass. For example a key to

unlock a door (what a strange use for a key!) could have "the small golden key" as its detailed description which is displayed when the key is found in a particular location or examined; its simple noun would be the word "key"; its location could be, let's say, the number four; and its weight may be one or two. These weights are usually only relative to other objects in the adventure and bear no resemblance to weight comparisons in the "real" world.

The inventory list is another feature that concerns objects. It is a list of all the objects that the player is carrying and the amount that is able to be carried can be restricted by two methods, should you wish to impose restrictions at all. Firstly, you could only allow a certain number of items to be carried before a message such as "You are carrying too much already!" is displayed.

Alternatively you could allow only a certain weight to be carried. In the former version, four objects could be the restriction whereas in reality four large and heavy objects are more difficult to carry than four small and light objects. It would be ridiculous to allow the player to carry four pieces of bulky furniture when if he tried carrying just five needles the message would be that he couldn't carry five because it is too much. It is therefore much more sensible, I think, to give each its own weight or "difficulty to carry" factor as mentioned in the previous paragraph. A cupboard could be given a value of one hundred and a book a value of four or five. For the locations of an object there is a simple method to use. Read on to find out!...

OBJECT LOCATIONS

Store them in arrays or actual memory locations. If a bottle of wine was in room five then it would initially have a value of five and an object "an empty bottle" would have a value of zero because it is not yet in the main bulk of the adventure. Room zero is an imaginary room and when the bottle is smashed, for example, "an empty bottle" would be given the value of the present room and "a bottle of wine" would be moved to the imaginary room zero. Then there is the problem that both bottles would more than likely have the same simple noun "bottle". To distinguish the two you must set a variable to zero if the bottle is intact or one if it is smashed, or vice versa. Otherwise, if you were to enter EXAMINE THE BOTTLE the program would not know whether it should display "the bottle is full of wine" or "the bottle is empty". You could also check to see which object number out of the two possible for the two types of bottle the player is carrying. The object number needs no further explanation as it is exactly what it says, just the number of the object in the whole list. Now you may say in the previous example that if you smash a bottle is no longer going to be called a bottle - in practice it becomes a pile of broken fragments of glass. I just wanted to provide a simple example - think of the wine being drunk rather than the bottle being smashed if it makes you feel better!

If something is in your possession then give it a value of 255 or -1. When the inventory list is displayed or when the adventure needs to check whether you are carrying an object, then simply check to see that the object's location number is 255 (or -1). That is all I plan to say about the pre-programming preparations for a while - on that subject we still have to cover the adventure's vocabulary. But now to a bit of programming.

NEXT MONTH

Remember that input command? No? Well re-read September's article first! Next month, although there will be no text, there will be an assortment of programs for you to try out in relation to this series. I am providing the text now so that you next time I can concentrate on the mega important side of everything! On that issue's disk you will find not only a further two pictures but also "AW-MODULES" and "AW-MODULES.MC". The former is a collection of routines that will be featured in the final adventure, Demad. The input routine will be there as well, together with parts that will "decompress" the picture files and display them, decompress the appropriate text and display it, and because a split screen is involved due to the graphics there is a routine to do just that - split the screen using rasters, and also there is a bit that will shift the bottom "text window" up without disturbing the colours of the pictures above. Just to round off, part of the machine code also clears the bitmap screen. Strictly speaking this is not how the final adventure program will be structured. At this stage I have kept everything as separate "modules" although the whole thing does RUN through completely as you may have expected or hoped. Whatever you enter in response to the input, the computer will at this stage do nothing. There is no routine there to analyse your command and none of the "verb" routines have been included or the routines to set

up variables.

If you just enter a number in response to the prompt, however, the computer will attempt to load that picture number, so long as the correct disk is in the drive. At this stage, though, when this happens the interrupts that provide the green bar at the top and the bitmap will be switched off whilst loading is taking place. A better effect will be produced a bit later, because the computer cannot actually access the drive and keep the interrupts looking as wonderful as they do at the same time. Having said that though someone is bound to say that it can be done, so I shall just cover myself by saying it cannot be done easily! You may, with a bit of programming experience behind you, be able to introduce impressive techniques such as sliding each picture in and out or clearing it slowly - literally bit by bit. I shall provide most probably the latter a little later in the series, maybe even in the next article as well if you're good. This will have to be in machine code because a BASIC version would take about half an hour! Just remember, you don't need graphics if you don't want them or don't feel confident with them. In the final adventure you will be able to switch off the graphics if you don't want them, but seeing as how my mate Doug (how are yer?) has done such an excellent job you won't want to switch them off will you!

BASIC HELP

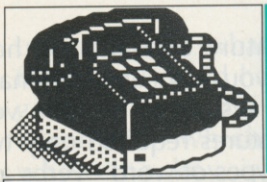
Also in the BASIC part of that program I shall provide a routine to read in the detailed object descriptions, the simple nouns, the initial locations and the weights of each of the objects in Demad. There are plenty of REM statements to help you understand each step and there is also a list of verbs and adjectives together with adverbs and linking words. More about these later on. The other important thing to consider is the direction information. At the moment this is all in BASIC and stored in various arrays. This method, although simple, uses quite a lot of memory and is not easy to manipulate if you want to do some things in machine code, which a lot of you won't anyway. If this is the case then the technique used here is ideal.

Each location needs its own set of eight numbers or bytes. These will represent what room is entered by moving north, northeast, ... and northwest from a specific location. If a value of zero is used then that exit cannot be used. For example if location number two was a north-south path with an eastern exit leading to room four, the eight number sequence may be 1,0,4,0,2,0,0,0. If you want the up/down option then simply give each location ten numbers and do similar if you only want the four principle directions - provide only four values. Compare the values in the program with the map that I provided in the last article. If a value of zero is given where you think there should be an exit available, it is due to an obstacle. Room eight, for example, requires a bridge to be repaired before the player can go south.

VOCABULARY

Back to the adventure's vocabulary. This need not contain a lot of words, let alone every word in the English language as a lot of people try to accommodate! The amount of words depends largely on what there is to do in the adventure and also on how complex you want the command analysing system to be. The very first adventures required only two different types of word - verbs and nouns. Inputs were very simple like TAKE AXE, EAT FOOD or TIE ROPE. Then they progressed slightly and the pronoun IT could be used to allow you to enter TAKE FOOD followed by EAT IT. The input routine simply assembles your command. This must be followed by a routine that splits it into individual words and determines what they are and what should be done. This is the most important and possibly, and invariably it turns out to be, the most complex part of an adventure and is called the "parser". The function it undertakes - analysing your input and deciding what to do as a result - is called parsing. It is, I suppose, a very simple form of artificial intelligence.

The complexity of commands that the parser recognises depends entirely upon your programming ability and the methods of analysis that you use. The production of a first class parser requires not only huge amounts of logical and rational thought, but that must also be combined with well-planned programming. So if you've simply been dying to write a super-wonderful state-of-the-art parser to beat all parsers, I may advise you to start saving for the pine-box, nails and flowers. But the topic of parsers will have to wait until next time. I shall provide then a BASIC parser to accept just the two word inputs like DRINK WATER, followed by, in the March issue a machine code one that will accept quite complex inputs. That is assuming I can work out how to program one!!! (September's issue included one such program called READING BETWEEN THE LINES, if you study this one, you will see the sort of things we will come up against).



CALLING ALL ADVENTURERS!!

Don't miss out on a golden opportunity to obtain another superb Graphic Adventure from that master author; TONY ROME, and at the same time stand a chance to win one of the following great prizes.

1st prize
An Amiga 500 computer

2nd prize
A Colour Monitor for your C64

3rd prize
A Colour Printer for your C64

4th prize
A selection of software from DIAMOND BYTES

5th prize
5 winners will each receive a copy of the GAC+ Adventure Creation System from INCENTIVE SOFTWARE

As well as the prizes opposite, every single person that completes the Adventure will receive completely free another great **TONY ROME** game called **THE SPANISH TREASURE.**

Mr. IAN ANDREW, Managing Director of **INCENTIVE SOFTWARE, DIAMOND BYTES** and **TONY ROME** have got together to bring you this special offer.

The **ARGON FACTOR** is set in the year 2155 and places you in the middle of the universe with only a broken spaceship and an android for company. Your job is to repair your spaceship and clear your name. You will travel to different centuries in your quest and all your powers of logic and observation will be needed to succeed.

```
2155 A.D.-SOMEWHERE IN SPACE your
ship is all that remains of a fleet of
Starships sent on a mission from Earth
to the limits of the Galaxy!..You are
Captain Cord a survivor of the Great
Zorvian War,now banished from Earth
accused of conspiracy.....
Your sole companion is Lap,a Venusian
Robot whose powers are now limited due
to damaged circuitry..You reflect on
those last moments with Anikra before
her capture by the treacherous Valdira
who had cleverly impersonated you with
a Halvian simulator.
Your only hope is to find the video
tape proving your innocence....After
crossing half of the Galaxy you have
your first lucky break! A faint
message from a strange planet.....
Your thoughts are broken by a sudden
movement in the ship's course..

   PRESS RETURN
```

Please send me.....Copies of THE ARGON FACTOR at £9.95 each
Please send me.....Copies of CODEMASTER at £15.95 each
Please send me.....Copies of POWER TOOLS at £14.95 each
(These programs are available on disk only)

I enclose a cheque/postal order for.....
(Made payable to DIAMOND BYTES)

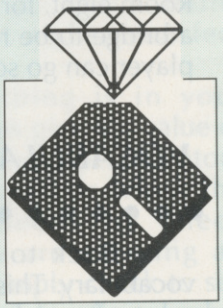
Name.....

Address.....

.....

.....Postcode.....

Send to: DIAMOND BYTES; 7 Graham Avenue, Brinsworth
Rotherham, South Yorkshire.
(Please allow 28 days for delivery)



The ARGON FACTOR will be available from 1st November 1990. Sole distributors are DIAMOND BYTES whose address you can find on the coupon opposite.

CONTRIBUTIONS

Written any good programs? Got some programming wisdon you want to pass on? Or do you want to write about your own field of interest? We're waiting to hear from you.

CDU doesn't just offer you the chance of appearing in print, but also of putting your program(s) on the disk for all to admire. We're always on the lookout for new programs for the disk and articles for the magazine. Anything goes, Utilities, Games or Business programs. They can be written in Basic, Machine Code, 'C' or whatever else you may write in. If we think it's good, we may well publish it.

Even if you haven't got a program to send, we'd love to pick your brains. If you have a field of expertise you'd like to explain or any hints and tips of interest to other disk users, send them in.

Just how do you go about preparing a submission? Just follow the guidelines and all should go well. You don't have to be a great novelist to contribute, but if you follow our simple rules then it will our job a lot easier. (and you stand more chance of having your masterpiece looked at).

1) First and foremost, ALL material sent in, whether it's a program on disk or just a written article, MUST have a SAE enclosed for return should you be unfortunate enough to have your submission rejected. Any such material received that does not include a SAE will NOT be returned.

2) If you do not own a word processing package then any text submitted with your material should be formatted as follows:-

A} All text should be double-spaced, i.e. there should be a blank line between each line of text. You should also leave a margin of at least 10 characters on each side of the text.

B} On the first page you should put the following information:

Name of the article/program.
Machine that it is for.
Any extras that are required - disk, printer etc.
Your name and address
Your telephone number

C} The top of every page should have the following information on it:

Abrreviation of the title.
Your name.
The page number.

For example, suppose you have submitted a piece on Extended Basic. You should put something like this at the head of each page:

EXBAS/F.Smith/Page...

3) On the bottom of each page you should put the word MORE if there are more pages, or ENDS if it is the last page.

4) Please do not put ANY additional marks on the text especially underlining. Do not use Italics/Bold/Subscripts etc etc to emphasize headings or sub-headings.

5) Try to write in clear concise English. Your submission does not have to be a masterpiece of the Queen's English, but it must be comprehensible.

6) Use a paperclip to hold the pages together, do not staple them.

7) Do NOT submit machine code programs as Basic loaders as in other magazines. However, if they make some specific point about codeing techniques, Assembler source files may be included.

8) Programs for the disk should be in as fewer chunks as possible.

CONTRIBUTIONS

This makes our disk menu easier to set up.

9) When submitting programs for the disk, submitting the program alone is not enough. Please tell us how to load; run and use it. If there are any interesting points involved, explain them to us.

BE WARNED, ANY PROGRAM THAT HAS NO SUPPORTING DOCUMENTATION WILL NOT EVEN BE LOOKED AT, no matter how good it is.

10) For people with word processors, a copy of the documentation in the form of a text file will be enough. In the case of WP files, do not put any special format commands etc etc into the text. Preferably, we would like any text files to be ASCII files. (If possible, PLEASE DO NOT use GeoWrite...It's a good program but causes us headaches). Some of the WP files we prefer are: Superscript, Easyscript, Paperclip II, Mini Office II or Amiga/IBM files.

11) If your article needs any artwork, then supply clear examples of what you want. We don't expect you to be an artist, but we do need to see what is required.

12) Photos, if necessary, must be either black and white prints or colour slides. Don't worry if you cannot provide photos, we can provide our own.

13) Payment for submissions can vary from 25 pounds for a very short routine to 800 pounds for a large program published in installments, and depends upon quite a number of factors, such as complexity and presentation of program, also originality. For written articles, the number of magazine pages is the salient factor

14) All payments are made in the month that the magazine containing your article has appeared in print.

15) If we do find your submission suitable for inclusion in the magazine, we will write to you giving the terms of publication, the rate of payment, and an agreement form. Prompt return of this form will allow us to use your program as soon as possible. (Though sometimes this can be quite a long time).

16) Send your programs and articles to:

CDU Submissions
Alphavite Publications Ltd
20, Potters Lane
Kiln Farm
Milton Keynes
MK11 3HF

17) CDU cannot accept any liability for items sent to the magazine.

NUMBER DEFENDER

Basic needn't be too restrictive when you want to write playable games as this offering shows

NATHAN GRAINGER

Everyone knows that to write a game of any credibility, you need to know machine code. Basic simply isn't capable of handling interrupts, sound control, multisprite movements and the other 1,000 or so things that good games need. However, after saying this, Basic is suitable for writing playable games that don't need all the attributes of the shoot'em ups. To this end, I have written for your enjoyment, quite a simple game that should still give you hours of enjoyment.

In the game, the player is put in charge of a shield, with the job of defending nine pillars, which are each numbered from one to nine. Bombs will fall (one at a time) from the top of the screen, in the direction of one of the pillars. It is then the player's job, by pushing a key on the keyboard (1 to 9) to defend that pillar. For example, if a bomb was about to fall into pillar number 7, the player would have to quickly press the '7' key on the keyboard. His shield would then move over the pillar labelled '7' and destroy the bomb.

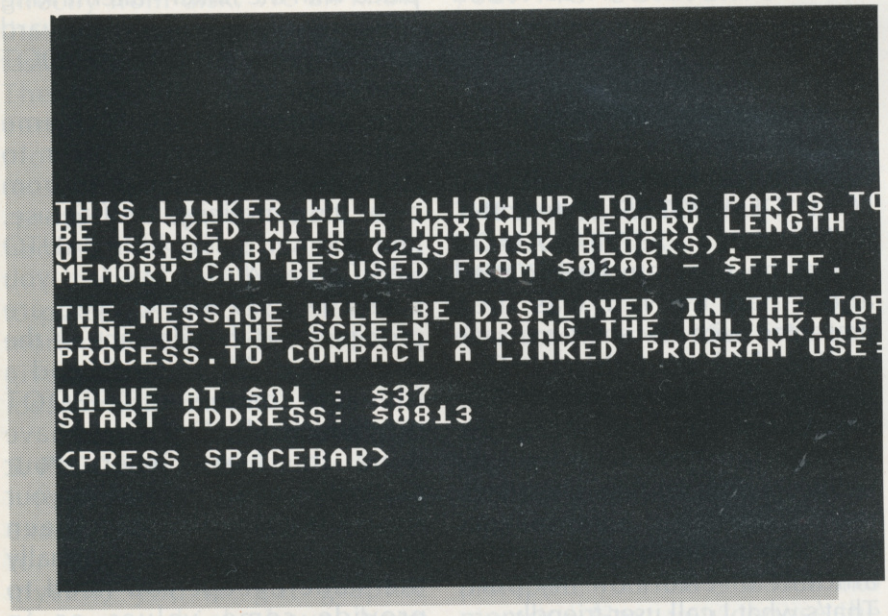
There are three levels in the game, with increasing numbers of bombs to stop on each. The bombs themselves are being dropped by aliens far above the planet, and if the player completes level 3, then he/she will destroy these aliens and thus win the game! However, if the player lets just one of the bombs hit any of the pillars, then it's 'game over'.

POWER TOOLS

DIAMOND n " a highly prized gemstone, the hardest of all minerals. A carbon cristallised in the cubic system: a rhombus:" A software house."

JOHN SIMPSON

After using CODEMASTER, a brand new utility, full of great possibilities and promise with fine new facets from DIAMOND BYTES, a unique, new software outfit which is currently making its mark in the world of utility software. I couldn't believe my luck when DIAMOND's latest package popped onto my mat with the instructions from the EDITOR of CDU, Paul Eves, to conduct an in-depth review of their latest offering for all erstwhile programmers, (experienced and the not so experienced). The package attracted my immediate interest after I had checked out the title...POWER TOOLS. Now there's something I'm always in need of.


THIS LINKER WILL ALLOW UP TO 16 PARTS TO
BE LINKED WITH A MAXIMUM MEMORY LENGTH
OF 63194 BYTES (249 DISK BLOCKS).
MEMORY CAN BE USED FROM $0200 - $FFFF.
THE MESSAGE WILL BE DISPLAYED IN THE TOP
LINE OF THE SCREEN DURING THE UNLINKING
PROCESS.TO COMPACT A LINKED PROGRAM USE
VALUE AT $01 : $37
START ADDRESS: $0813
<PRESS SPACEBAR>

NICE INTRO

Okay then, I thought, let's wack on the power and spin the diskerooni. The first thing that struck me neatly between the eyes and ears - like a bullet from the gun of a `man with no name' - after a fast and colourful load is the great menu screen. Scrolling, vibrant technicolour, much like you would get with game software, and not what you usually expect with such a serious utility. Add to this a rather neat multi-track musical score, with a sine-wavey scrolling message line, and, if your're like myself, you'll probably find yourself not doing too much for quite a while, just simply drifting off whilst listening to the music and having a quiet medative moment or two.

MEDITATION OVER - WORK TO DO

Six options are highlighted on the menu which are:

1. LINKER
2. AUTOBOOT
3. COMPACTOR
4. SCRAMBLER
5. PIC CONVERTER
6. SCROLL WRITER

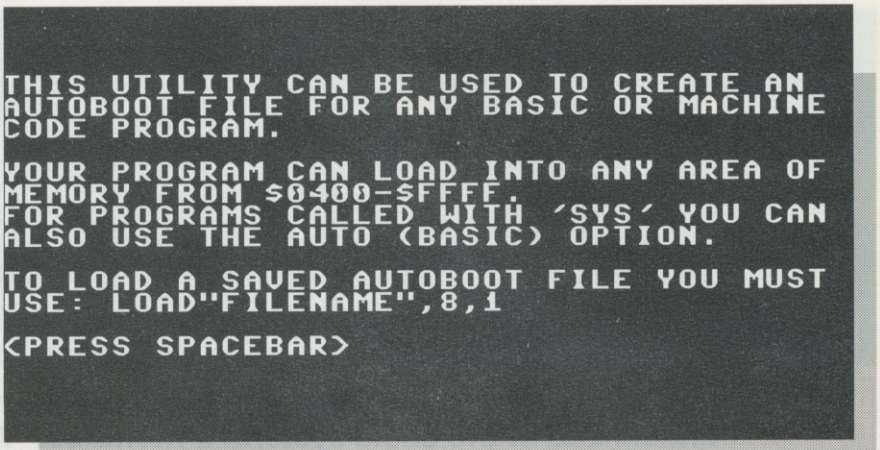
Right, let's look at each option in more detail, although as you can see from the menu names, they are pretty much self-explanatory. In fact, that is the beauty of this package, self-explantory, and simplicity itself to use. But then again, isn't that the essence of what a good package should be?

1. THE LINKER

This little gem from the suite of six programs is just the job for joining various files which, together, comprises a complete machine code program. Now I've got a neat utility that I've been working on for some time and, which consists of a couple of code files in low memory, a character set at $2000, a group of sprite data at $3600, and a whole bunch of screen data from $4000 upwards, plus - right out on a limb - another code file which resides at $C000. Could I link all this together? I thought.

No problem!! Pressing the function key F3 allowed me to type in a "message", a neat facility to allow the emplacement of the program name, author, or some other message, which is displayed on the top line of the screen whilst the LINKER is unpacking the files back to their origination.

To select the actual linking process press F5. Here I found that I could link up to 16 files giving a maximum program length of some 63K bytes. Phew! who writes programs that long? (People with lots of screen and mucical data John...Ed!!). Each file must load into memory between 512 and 65535 ($0200-$FFFF). That will give you virtually the whole of memory. Pretty POWERful stuff what!!


THIS UTILITY CAN BE USED TO CREATE AN AUTOBOOT FILE FOR ANY BASIC OR MACHINE CODE PROGRAM.
YOUR PROGRAM CAN LOAD INTO ANY AREA OF MEMORY FROM $0400-$FFFF.
FOR PROGRAMS CALLED WITH 'SYS' YOU CAN ALSO USE THE AUTO (BASIC) OPTION.
TO LOAD A SAVED AUTOBOOT FILE YOU MUST USE: LOAD"FILENAME",8,1
<PRESS SPACEBAR>

LINKING

All the files I needed to link were all on the same disk in the drive, so off I went. "Strike key F5". I was immediately presented with another screen giving me the first disk directory entry and a flashing cursor over YES/NO. I selected Y for yes, and was straight away presented with the next file on the disk. Press Y or N until all the files you want to link are selected. Pressing RUN/STOP aborts and RETURN skips the rest of the directory.

Once I had completed the selections I was then prompted for a value (2 HEX digits), to be loaded into memory location $01. (As we know, this location controls the memory configuration), very important if the program's RUN address happens to be sitting under ROM. Mine wasn't, so I simply typed in $37 (standard power up condition), and was then prompted for the program's RUN address (A 4 digit HEX number). Finally, a further prompt to enter a file name for the linked program files.

A few minutes wait...well, I did link an awful lot of info, and then I was asked to insert a final destination disk. (Nice touch that). The linker then saved out my final program. The computer neatly reset itself, and now I could test the worthiness of the link by using a normal load command from Basic:

LOAD"DEMOPROG",8 <return>
RUN <return>

Once the program had loaded and I had RUN, I was then presented with my short message which I had created from "F3 - Edit Message", whilst the LINKER unpacked everything. When the job had finished, there was my utility, up and running. The whole process taking just a few minutes from start to finish with the least amount of effort on my part. That's what I call user-friendly.

I'm going to give the LINKER 9 out of 10. The only reason it doesn't get 10 is because I felft that when loading the file info to be linked, it was a little too automatic. That is to say, I would have liked to option to type in my own file names, and exit the input with a keypress, say a function key. Why? Well, if you're like me, and you've got a large number of files clumped onto one disk, it can take some time skipping through them all until you find the ones you want....still, this is such a personal and minor quibble that I'll up that value to 9.7 out of 10.

2. AUTOBOOT

This program slips nicely into place with the linker - after linking it's certainly a straight forward enough process to AUTOBOOT your program of LINKed files.

The program is as the name sugest, and allows a user to AUTOBOOT any program from disk. It's simplicity itself to use, and you can select from "AUTO BASIC" or "AUTO M/C". If you select "AUTO BASIC" you are prompted for the filename of the program to be booted, and a filename for the AUTOBOOT program itself. Once you have done this, you insert your destination disk, in seconds your AUTOBOOT file has been created. "AUTO M/C" is virtually the same except you need to provide some values as in LINKER.

3. COMPACTOR

This, the third program of the suite, is a very neat compression type of utility. Perfect for files containing repeated bytes, such as sprite data, character data etc. Once again, simplicity itself to use. Just type in the source file name, then a destination name, next the value at $01, and a start address. Bingo!! Even Basic programs can be compacted using this facility. Just enter $37 at "VALUE in $01" and $A7AE at "START ADDRESS"

```
THIS UTILITY WILL HELP PROTECT YOUR
PROGRAMS FROM BEING HACKED DIRECTLY
OFF DISK.

THE SCRAMBLE CODE IS SIMPLY USED AS A
VALUE TO EXCLUSIVE-OR THE WHOLEFILE.

ALL FILES TO BE SCRAMBLED MUST
ORIGINALLY LOAD AT 2049 ($0801).
```

4. SCRAMBLER

Another little gem! This program will help protect your programs from being hacked directly from disk. We all know that it is a decidedly impossible task to offer absolute protection. There's always someone who can hack into anything that moves, but this does offer reasonable

protection against the casual hacker. I'll say no more about this routine, you'll simply have to purchase the disk to find out more...I certainly don't want to give any would-be hacker any ideas or information.

5. PIC CONVERTOR

I found this program to be exactly what I've been waiting for for a long time. If, like myself, you've got graphic pictures which have been created using various editors, and you want to convert them to edit within a different editor, then in the words of a famous commercial, this is the one for you. PIC CONVERTOR can cope with six such graphic packages:

THE ADVANCED ART STUDIO
ARTIST 64
BLAZING PADDLES
THE IMAGE SYSTEM
KOALA PAINTER
VIDCOM 64

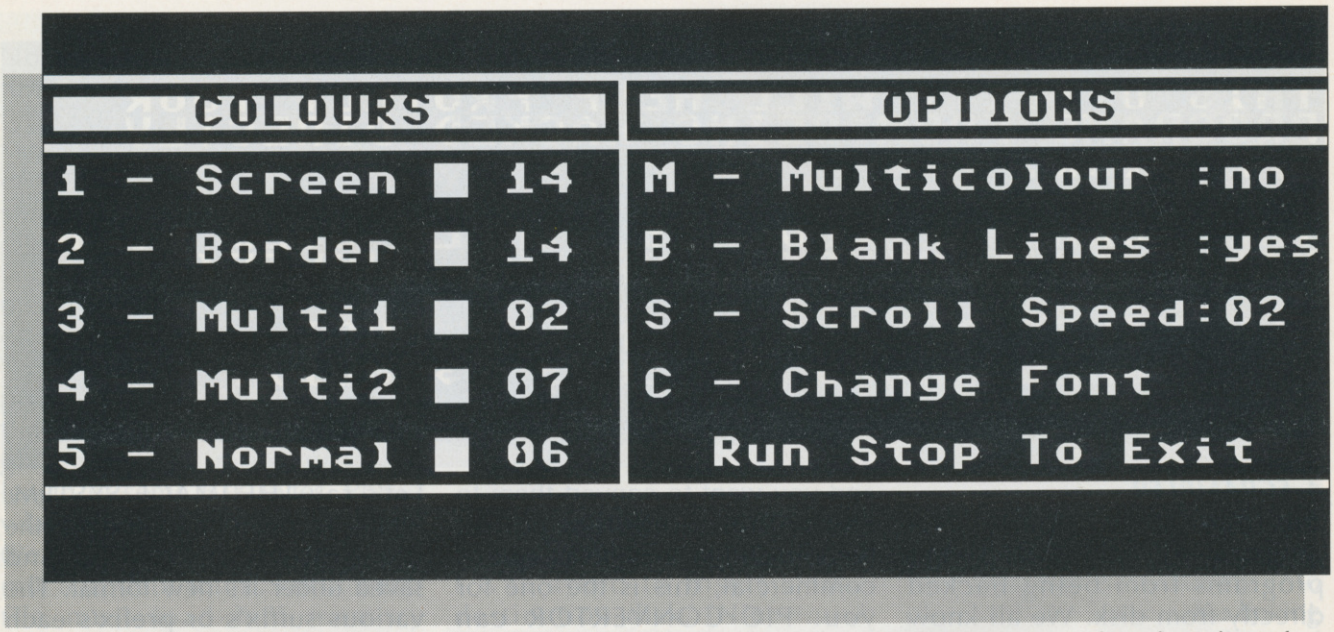
The screen is divided into two windows, the SOURCE and the DESTINATION. Each window lists the graphic packages mentioned above.
First select the source picture file. Lets say one from ARTIST 64. Using the function keys as prompted, your file is loaded into memory. Next you choose the format you wish to convert it to. Let's say THE IMAGE SYSTEM. Again, following the prompts, the file is now CONVERTED and saved under it's new format. The various suffix's or prefix's each editor may use are automatically taken care of. This program is great for work you might have done earlier and now need to convert to your new system.

6. SCROLL WRITER

The final program of the suite, and what a finale it gives to us. I have two speeches when I come across utilities such as these. A short speech and a long speech. The short one is: Excellent. The long one is: B....y Excellent.

There are ten items on the menu to select form. Selecting the EDIT/VIEW TEXT option creates a `double' window, the left side displaying the various commands such as; centered, left-aligned,

```
SOURCE                    DESTINATION
ADV.ART STUDIO            ADV.ART STUDIO
ARTIST 64                 ARTIST 64
BLAZING PADDLES           BLAZING PADDLES
THE IMAGE SYSTEM          THE IMAGE SYSTEM
KOALA PAINTER             KOALA PAINTER
VIDCOM 64                 VIDCOM 64
```



wrap-around, text colour, clear, goto end of text, start of text, next line etc etc. The right window is the actual text editing window. Once you have created some text, you can TEST SCROLL it to see just exactly what it will look like.
There is a FONT/COLOURS option which allows you to change screen, border, character colours, switch on/off multicolour, set the scrolling speed and even change the font itself. There are four fonts, built in, to choose from.

A) Standard style
B) Bas-Relief style
C) Designer style
D) Computer style

It doesn't stop there however. You can design your own fonts (the characters need to be 2 by 2 in size), using a good character editor, such as the new UDG SYSTEM 2 character and screen editor. (A review of which will be shown in the next issue). When you have designed your text and everything is ready, you can save this out as a working demo. It will use 33 disk blocks unless you include music. Talking of which, almost any music routine can be played from within your demo, under the following restrictions;

1. It must be IRQ driven
2. It must not use locations $0400 to $27FF
3. Max program size is 18432 bytes
4. You must supply an address in hex for;
   a) Inialise music
   b) IRQ music play

As you can see these restrictions are not too severe, and you still get 19K for a program. Of course, you don't have to use this music track.
This last program, "SCROLL WRITER", from the suite of POWER TOOLS is a little beauty, allowing you to save and load text, load new fonts, use all the disk commands, as well as Edit/View text, test scroll and create a demo.

IN CONCLUSION

Each program is fully independant of the rest, and each program has its own menu system. Also, each menu system has a `Help' call to information, just in case you might forget something. All the programs are easy to use with the minum of fuss `n' bother, and Author/Software-house ego trips don't get in the way of things.
All in all, I must say that POWER TOOLS in the hands of a programmer is a very valuable addition to a library of utilities. Without which an empty void must echo a lament for the six missing programs of the suite. All credit must go to the author, NM156 (Is it an Android I ask myself), for such good work. Also to DIAMOND BYTES for recognition of a suite of utilities that most other software houses could easily have overlooked in their never ending quest for the `ultimate game'.
My final recommendations are this; If you are a programmer and you can spare the few pounds this delightful package costs, go and get it.

PROGRAM:	**POWER TOOLS**
PRICE:	**£14.95 (DISK ONLY)**
SUPPLIED:	**DIMOND BYTES, 7 GRAHAM AVENUE, BRINSWORTH, ROTHERHAM, S.YORKSHIRE**
RELEASE DATE:	**IST NOVEMBER 1990**

SEE COUPON ON PAGE 12

MEMORY SCANNER

TAKE A LOOK INSIDE YOUR COMPUTER'S MEMORY WITH THIS EASY TO USE BASIC PROGRAM

LEE BAMBER

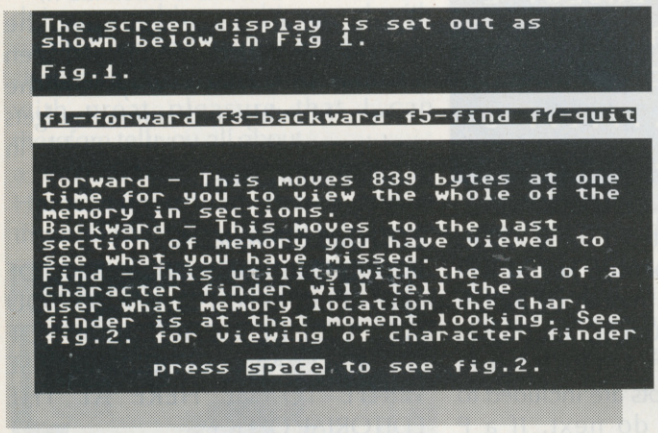
Scanning the memory has been such an easy task for the gifted programmers of our age, and yet so easy that no-one attempted make a simple program based on the idea. Now you have the chance to see exactly what those programmers where viewing all this time.

The MEMORY SCANNER is a simple Basic program which shows you every byte in the whole of the computers memory which can be used to your advantage. An introduction has been produced on the disk to show you what the utility is capable of, and explain the utility commands one at a time. If you wish to see the program and how it works, it is easily listed by quitting the program.

BIRTH OF A PROGRAM

The idea came to me when I was playing an Adventure and had come to a fairly difficult part. I was stuck at a particular puzzle for weeks. I therefore decided the only answer was to have a look inside the computer's memory and find the necessary keywords needed. I reset the computer and used my original little Basic program. To my surprise I found an entire range of Objects, People and Vocabulary in a certain block of memory. After improving my simple program a little, I ironed out some of the minor bugs and **'MEMORY SCANNER'** was born.

Below is a quick explanation of the four instructions needed.


The screen display is set out as shown below in Fig 1.
Fig.1.
f1-forward f3-backward f5-find f7-quit
Forward - This moves 839 bytes at one time for you to view the whole of the memory in sections.
Backward - This moves to the last section of memory you have viewed to see what you have missed.
Find - This utility with the aid of a character finder will tell the user what memory location the char. finder is at that moment looking. See fig.2. for viewing of character finder
press space to see fig.2.

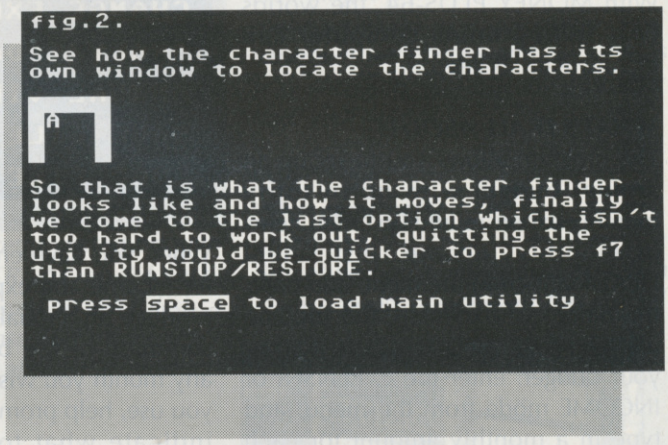


fig.2.
See how the character finder has its own window to locate the characters.
So that is what the character finder looks like and how it moves, finally we come to the last option which isn't too hard to work out, quitting the utility would be quicker to press f7 than RUNSTOP/RESTORE.
press space to load main utility

FORWARD - This command will show the proceeding block of memory. The block of memory being 840 bytes long. Note that when you reach memory location 65280 (decimal) the FORWARD command will not work.

BACKWARD - This command will view the previous block of memory which the user had seen last. Obviously, when you reach location Zero, this command will not work.

FIND - This command will activate a character finder which will enable you to move a pointer around the screen using a joystick in Port 2. The characters position in memory will appear as a flashing number at the top of the screen below the options menu. This option can be escaped by pressing the fire button which will return you to the main menu.

QUIT - This command will simply end the program and enable you to list the prgram and perhaps change it.

SOME USES

The **MEMORY SCANNER** can be used for a variety of reasons. Take for instance if you are playing an Adventure and wanted to see what objects you needed to collect. You simply reset the computer and load and run **MEMORY SCANNER.** You would eventually come across a selection of words or objects which you simply jot down then when you play the Adventure again, you would know what you were looking for.

It could also be used for finding those cheats in games, by looking through the memory and finding words that are not used in the game itself. If you wanted to impress your friends, you could find the High Score table and alter it to suit yourself. There are numerous reasons why you might like to have a look at the memory, the **MEMORY SCANNER** makes it that much easier.

The only real drawback is that because it is in Basic, and therefore always sits in the same place in memory, the computer's memory from the start of Basic until the end of the program is obviously always the same. (IE: The program itself).

I hope you enjoy using this simple utility as much as I have, for it supplies you with an easy way of hacking and some extended information about the inside of your computer.

MONEY PLUS-64

FINANCIAL PLANNING IS A MUST FOR THE 90's. Let this program make the task that much easier

PETER SIMMONDS

This is a Commodore-64 version of a very useful financial planning program that is already used by many Plus/4 and C128 owners. Originally published for Plus/4 users in a `YOUR COMMODORE' supplement, a revised version was included on the April 1990 CDU disk. With 1990 being the year of the `Financial Pinch', sound planning of financial affairs has become more fashionable. With MONEY PLUS-64, the worlds best selling home computer can literally earn its keep.

WHAT'S IT ALL ABOUT?

The program allows you to set up an annual budget with up to thirty monthly expenses and up to ten items of monthly income. Just select HEADINGS from the menu and enter the names of expenses and income using up to twelve characters. To start your budget, enter the EXPENSES or INCOME mode from the menu, and type in a monthly amount for each heading. CURSOR keys select the headings and months. Use I or E to switch between entering income items and expenses. You can only budget to the nearest pound and only up to 9999 pounds for any single item. Use RETURN to enter one item; using A will cause all twelve months to have the same value.

```
MENU
CALCULATE..... C
CHANGE HEADINGS. H
MONTHLY DISPLAY. M
PRINTER..... P
CHANGE INCOME... I
CHANGE EXPENSES. E
ANALYZE DATA.... A
ANNUAL VIEW..... Y
SAVE FILE..... S
LOAD FILE..... L
```

When you have entered some values for income and expenses, key M will produce an annual, month by month summary of the budget, starting at any month you wish. On each screen you use, help prompts are included to indicate what to do next. If a P appears in the list of prompts, keying P will take you direct to the printing mode. You can print a full annual budget or a single months budget. Other prompts work in the same way to by-pass the main menu.

ERROR TRAPS

The program is interlocked to stop you doing silly things. For example you cannot print a budget until you have created one, together with its numerical information. To print a budget, any 80 column printer should do. The MPS801 and DPS1101 are O.K. Other printers with a suitable interface should not present a problem. A PLUS/4 user in AUSTRALIA has a Citizen 128D and has had no problems whatsoever. (I can confirm that this is true....ED!!)
If you do not have a printer, the budget can still be set up and then viewed in various ways on the screen. Monthly and annual budgets and item analysis are all available in a user friendly style.

EDITORS COMMENT...

IF YOU DO NOT HAVE A COPY OF THE APRIL 1990 ISSUE, AND YOU WOULD LIKE A PHOTOCOPY OF THE RELEVANT ARTICLE, PLEASE WRITE TO ME HERE AT THE EDITORIAL OFFICE.

WE HAVE PROVIDED TWO VERSIONS OF THE PROGRAM ON THE DISK FOR YOU. THE FIRST IS A COMPILED VERSION FOR THE EXTRA SPEED. THE SECOND, NAMED MONEY PLUS.BAS, IS SO THAT YOU CAN ALTER IT TO YOUR OWN NEEDS.

```
INCOME
<LEFT>   MONTH  JUL   <RIGHT>
<UP>     ROW 1  SALARY   <DOWN>
         1000
1 TO CHANGE PRESS [C]
2 USE RETURN TO ENTER
3 CHANGE ALL YEAR [A]
4 TO QUIT PRESS [M]
```

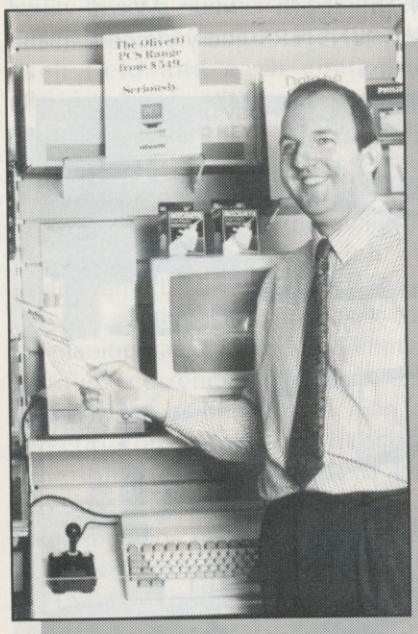
```
ANNUAL STATEMENT
     CREDIT  DEBIT  BALANCE  ACCUM.
JUL    1072    977       95      95
AUG    1058    972       86     181
SEP    1058    922      136     317
OCT    1072    932      140     457
NOV    1058   1037       21     478
DEC    1058   1028       30     508
JAN    1072    978       94     602
FEB    1078    800      278     880
MAR    1078    865      213    1093
APR    1012    853      159    1252
MAY    1058    887      171    1423
JUN    1058    882      176    1599
      12732  11133     1599    1599
(PRESS M.P.E OR I)
```

SERVING THE PUBLIC

In the world of entertainment and business solutions it's nice to see there are some companies not only ' Serving the Public', but also expanding because of their dedication

S. WICKHAM

These days, it seems to be the `in' thing to knock others for our own failures and shortcomings. All too often I hear things like; It's not my fault, we were let down by the suppliers. Or, I blame M..... T....., it's all her fault. Etc, etc. Therefore, when you come across someone thats doing something positive, and with success, you feel that the whole world should share in your discovery. It is with great pleasure that I can therefore tell you all about a local computer company, based in Hemel Hempstead, that is one of those `SERVING THE PUBLIC'.



The proprietor of Faxminsters Mr. Ward showing his wares.

Faxminster computers of Hemel Hempstead have been trading for some 6 years now in the field of computer hardware and software. I popped down to Hemel last week and spoke to Mr. Noel Ward, Managing Director and founder of this small, but successful company.

SW
Tell me Mr. Ward, what got you started in the first place?

My roots lie in computer sales and field management in the business sector of the industry. However, back in the early 80's, I could see the opportunity that was avaliable in the retail/leisure market. I started off with the BBC/SPECTRUM and LYNX. Initially, I managed to establish a good customer base and made sure I knew something about the products that I was selling.

SW
The mid 80's saw a lot of small businesses disappear from the high streets. How did you manage to stay afloat?

At the point of the big cut-throat wars, about 3 years ago, I saw the dangers and avoided the traps that so many others fell into by continueing to produce an already consolidated customer base. I also had the foresight to realise that to provide a wider and better service, I had to concentrate on providing not just the computer, but the peripherals and accessories to keep the computer serviced. In other words I provided SUPPORT.

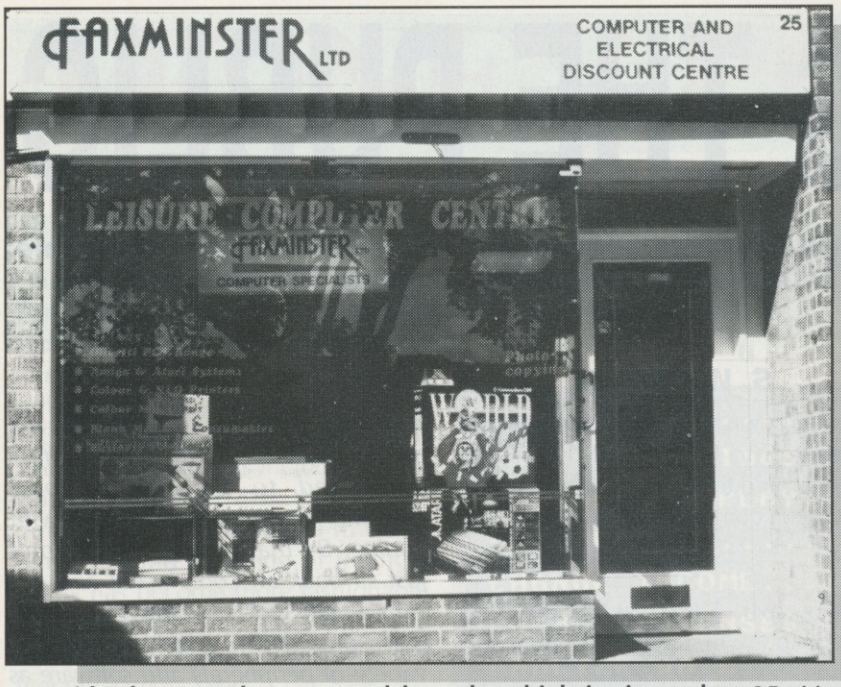
SW
Thats fine Mr. Ward, but how did you manage to expand once you had become established?

With the advent of the 16 bit revolution, business really took off. I continued to provide a good customer support service along with choice. By that I mean that I didn't just sit back all complacent and say "I'm alright Jack". I wanted to ensure that I provided as much choice and variety of peripherals and software as was necessary to keep my customer base well serviced.

SW
What came next?

Since we had managed to establish such a good business in the Hemel

A game players dream come true??



Outside the Hemel Hempstead branch, which is situated at 25, Market Square, Hemel Hempstead. Telephone : 0442-55044 or FAX: 0442-219607

Hempstead area, we wanted to expand into areas where there was a need to satisfy the demand of the computer user. Dunstable seemed to ideal place due to the number of surrounding villages.

SW
Obviously competition from the big boys must always be like a sceptre hovering over your shoulder. How do you deal with this?

The simple answer to that is this. The big boys, as you call them, are in reality a help to us. As you know, most of the high street stores only sell the computer's. Through our support service, we can keep all their thousands of customers happy. They may sell the computer's to their customers. But we supply the peripherals, software, back-up and support. What more could we ask?

SW
What do you see of the future?

We shall continue to provide the kind of service that we have become famous for. The underlying factor though is that we are constantly keeping our eye on the market and the market trends. Because of this understanding of the industry we can provide the flexibility that is demanded of us.

SW
I would like to thank you Mr. Ward for taking the time to talk to me. I wish you every success for the future.

Thank you very much for giving me the opportunity of speaking, through your pages, to the many thousands of computer users out there that sometimes need reassuring that somewhere, there is someone that understands their needs.

EDITORS COMMENT

I would just like to add my own personal observations to the above commentry. I have been dealing with Mr. Ward, and FAXMINSTERS since 1984. (Long before I got into the field of magazine publishing). I can honestly say that I have always found him to be both knowledgable and helpful. It's all too easy these days for companies to sit back and do nothing. (See the article WORTH THE HASSLE). I would also like to add my thanks, from a professional point of view, to FAXMINSTERS for providing us with a lot of the software/hardware that we review in CDU and our sister magazines YOUR AMIGA and YC....I wish both Mr. Ward and FAXMINSTERS every success in the future.

At 18 Church Street, Dunstable can be seen the smart new interior of Mr. Wards second shop. Telephone: 0582-475474 FAX: 0582-475455

xinout

A Basic extension primarily concerned with Input, Output and Data Handling

C.P. McELHINNEY

XINOUT provides the programmer with the most powerful INPUT command available on any micro, (Quite a strong claim....Ed!!), with the ability to set a number of parameters describing the length and type of each field and each character, allowing cursor movement between fields and input pages, and with on-line help.

Output is improved by the inclusion of PRINT USING, which is implemented as a command and a function, so the actual data itself can be maniupulated (rounded, centred, justified, currency formatted etc.), and not just its image on screen. A selection of useful string-handling functions are also included. The program takes nothing away from Basic memory, making extensive use of shadow RAM and RAM at $C000 (49152) upwards.

EXTENDED COMMANDS

XINOUT includes the following enhancements and extensions:-

INPUT - Enhanced to include parameters and allow variable prompts; also made recursive (i.e INPUT prompt,variable;prompt, variable etc.); a comma after the prompt suppresses the "?"; LINPUT and LINPUT# for line input;INPUT @ (see the RAIL ROAD diagram for all variations).

```
Forename :Eves          :
Surname  :Paul Antony       :  Sex   :1:

Street   :104 High Street       :
Town     :Berkhamsted       :
Postcode :HP4 2BL:

D.O.B    :14/08/50:

Height   :6.01:          Weight   :10.10:

Salary   :       :

NI number:YM 35 34       :

Notes-
:The Above details are not necessarily
 accurate but they serve to give a demo:



? for help:CTRL N(ext):CTRL P(revious)
```

You may only have a comma if using LINPUT or a number of variables (e.g INPUT A,B,C). A beep is sounded if an error occurs; this can be disabled/enabled by toggling CTRL-G. CTRL-E will move the cursor to the end of the field.

The Kernal input routine has been re-written so that UP and DOWN cursor keys, CLR, HOME, CTRL C,N,P or Z, or filling the field will terminate entry, you can't delete past the start of the field. RUN/STOP is ignored in INPUT mode, you can gave at most 1 decimal point and one `E' (for scientific notation) in numeric fileds.

RUN - Enhanced to include `RUN filename$,device' to `CHAIN' a program. As a by-product of re-writing the input routine, RUN/STOP will load and run the first program from disk, by forcing

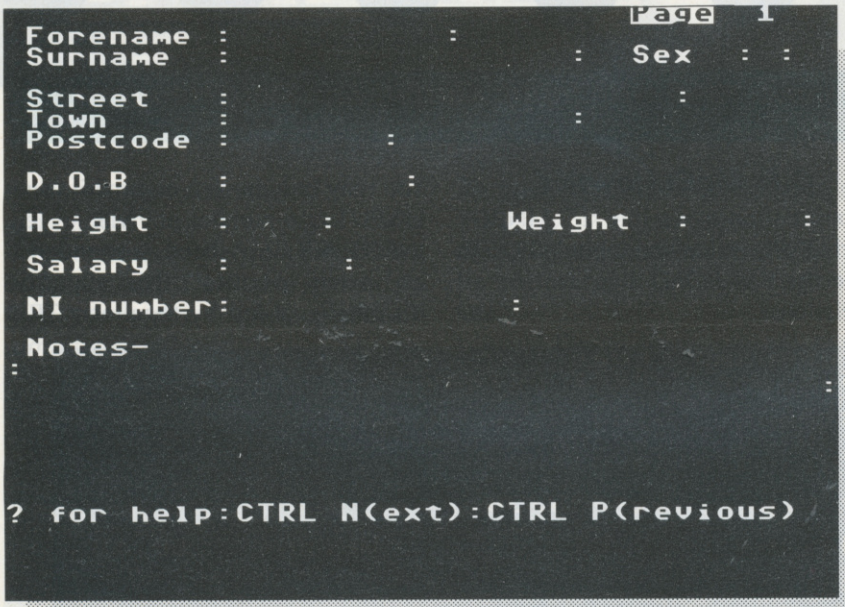
ON THE DISK

'rU"0:*",8<CR>' into the keyboard buffer. This can be changed by the RUN$ command. E.G RUN$="load <CR> run <CR>"; note that the back-arrow can be used for <CR> and apostrophe for quotes.

RESTORE - Includes RESTORE linenumber.

USR - Allows PRINT USING command as on the C128.
E.G USR "###.##";1.346,2,21.9
The above would display; 1.35 2.00 21.90

DEF - Same as Basic 7 PUDEF command.



PRINT - Allows PRINT @X,Y [print list] equivalent to BBC PRINT TAB(X,Y).

IF - Modified to include new commands; many extensions require a colon after THEN if using new commands. Also allows BBC type IF condition action structure, no need for THEN.
E.G IF AGE < 5 @ 0,0 "NO CHARGE"

@ - Used to position the cursor:-
@ x,y (x coordinate (0-39) y coordinate (0-24))
@ x,y PRINT printlist e.g @ 30,0 PRINT"Cost =";cst
@ x,y INPUT e.g @ 0,0 INPUT"Press return",
@ x,y LINPUT
@ x,y print list e.g @ x,y+2val(a$);" pounds"

INPUT PARAMETERS
Input parameters can be set globally if they occur on their own, for example:-
&LEN(8) VAL"9".0
would set all subsequent inputs to 8 characters of type "9" with no decimal places. Alternatively they can be set locally, only affecting the current input, if they occur in the input statement itself, after any prompts and before the comma or semi-colon. See the RAILROAD diagrams for the INPUT and input parameters.

CLR - Sets all input parameters to default and clears &FOR stack. Include at the start of all programs.

END - Sets list of keys which will terminate input. Default is ? and STOP. The functions &END and &END$ can then be sued to carry out some action based on the terminator. For example; IF &END$="!" THEN... Note, does not affect fixed terminators i.e. CURSOR UP, DOWN, HOME, CLR< <CR>, sh-<CR>, CTRL N/P/Z, field full.

USR - Sets `INPUT USING' mask. Defines type of each character in field, and the length of the field. Types are as with VAL, with the addition of type `#' which means don't care. i.e no type checking; also, any character in the mask which is not a type will replace the character typed at the position.
For example; &USR"Aaa 99/99/99" would be a suitable mask for input of the type Day DD/MM/YY, i.e Mon 01.01.90.

VERIFY - Sets alternative mask, specifying the range of each character, for example, a mask of "AZ09" would set field length to 2, first character between A and Z, the second character between 0 and 9 inclusive.

LEN - Sets field length, if the field is filled, input is terminated, returning zero in &END function. LEN 0 resets field length to default, which is two screen lines.

VAL - INPUT TYPE. Sets the field type.

9 - Numeric only (upper/lowercase 0 and 1 mapped to 0 and 1shifted numerics mapped to numerics).

x - Alphanumeric upper/lower case.

X - Alphanumeric upper case.

a - Alphabetic upper/lower case.

A - Alphabetic upper case.

S - Sign. + or -. (Shifted +/- mapped to +/-).

L - Logical. 0,1,t,T,f,F,y,Y,n,N.

INT - INPUT INSTR. Sets a list of valid keys for the input field. For example;
&INT"0123456789ABCDEF"
would allow only hexadecimal input. A useful addition is the ability to set `inclusive' and `exclusive' INSTR's; for example;
&VAL"9" INT +"ABCDEF" would allow hesadecimal input with advantage of type "9", i.e; shifted numerics mapped to numerics etc.

&VAL"A" INT -"AEIOU" would allow only consonants.

**LEFT$** - Sets whether leading spaces on the left have to be truncated. &LEFT$ 0 means do not truncate; &LEFT$ 1 means do.

RIGHT$ - Sets whether trailing spaces have to be truncated.

OTHER AMPER COMMANDS

There are also the commands &FOR, &NEXT, &H, &LOAD and &SAVE associated with INPUT although not as input parameters.

&FOR - This allows easy input to a one or two dimensional array, used along with the functions &P, the page counter and &I, the input counter.

&NEXT

```
For example;
&FOR P=1TO20
&FOR I=1TO10
@ X(&I),Y(&I) INPUT
FIELD$(&P,&I)
&NEXTI
&NEXTP
```

The &NEXTI command will update the input counter depending on how the field is terminated, allowing the use of the cursor keys to move up and down between fields (with wraparound), <RETURN> or sh-<RETURN) to move on to the next field, or on to the next page if all the fields are filled. The ? will display the input help screen, CLR blanks the current field, HOME moves the cursor back to the start of the field. CTRL-N and CTRL-P will terminate the &NEXTI loop. &NEXTP will move on to the next page if the inner (&I) loop is terminated by filling all fields or <RETURN>, back a page if terminated by CTRL-P (displaying first page if no previous), and forward a page if CTRL-N (displaying `last page' if no next page).

&FOR and &NEXT can also be used with counters &X and &Y which provide a slightly faster FOR...NEXT loop, useful for reading in INPUT MASK data or clearing an array etc.

&H - Displays the help screen, waits for a RETURN then re-displays the last screen.

&SAVE - Saves the current screen in shadow RAM.

&LOAD - Reloads the saved screen from shadow RAM.

FUNCTIONS

&. - Returns the current number of decimalplaces in numeric field, as set by for example, &VAL"9".5

&C - Returns number of characters output since last <CR>. Equivalent to the BBC COUNT function.

&O - Returns number of characters output. Not set to zero by <CR>.

**&L$** - LOWER$ function; converts upper case letters to lower case; e.g PRINT &L$("COMMODORE C64") will return "commodore c64".

**&U$** - UPPER$ function.

**&END/$** - Returns ASCII code of character which terminated input (&END), or the character itself (&END$). If field is terminated by being filled, returns 0.

&POS- INSTR function; e.g. &POS(A$,B$[,x]). Returns position in A$ at which B$ occurs, optionally starting from xth position. Returns zero if B$ does not occur in A$.

&VERIFY$ - Returns current input mask string. (See &VERIFY parameter).

&SPC - SPACE$ function. Returns string of spaces. Useful for initialising fields or clearing fields.

&INT$- Returns current INPUT 'in string'; list of valid keys in field, as set by &INT parameter.

&USR$ - (1) Returns the current USING mask as set by the USR parameter.

(2) USING$ function. All the features of BASIC 7 PRINT USING command, but implemented as a function, so that operations can be carried out on the data itself, rather than just on its screen image.

Syntax; &USR$(mask$,item,item..., item)

Example; HEADING$=USR$ ("#####",1,2,3,4,5) would return; " 1 2 3 4 5" in HEADING$.

See also the USR command.

&INPUTX - Returns x coordinate of start of current input.

&INPUTY - Returns y coordinate of start of current input.

&LEN - Returns length of current input. Zeor if no length specified.

**&STR$**-STRINGfunction. &STR$(length,char[,/STEP incl]). Returns a string of `length' chars, optionally incremented by `inc'.

Examples:

PRINT &STR$(10,"-") would display

LINE$=&STR$(30,63) would return "??????????????????????????????" in LINE$

&INT &STR$(26,"a",1) would set the INSTR parameter to lowercase alphabet.

PRINT &STR$(5,"0" STEP 2) would display 02468

**&VAL/$** - &VAL returns the ordinal value of the current INPUT TYPE. &VAL$ returns the type ("9","x","X","a","A","s" or "1"); if USING mask is specified (see &USR parameter) the mask is returned. If &VERIFY mask is specified, the &VERIFY mask is returned.

&LEFT$ - Returns 1 if left truncation set, 0 otherwise.

&RIGHT$ - Returns 1 if right truncation set, 0 otherwise.

&X, &Y

&I, &P - Returns the value of the &FOR loop counter.

SPECIAL VARIABLE - IN$

The variable IN$ is cleared at the start of each INPUT and a copy of the actual entry is assigned to IN$ at the end of each INPUT. This is useful to ensure that data is entered. e.g;

```
10 @ 5,10 INPUT "NAME ",NAME$

20 IF IN$="" @ 0,0 INPUT"Field must be present",;GOTO10
```

ON THE DISK

STAR COMMANDS

These commands evolved as by products of re-writing the output routines.

***S** - Equivalent to Apple SPEED command. Sets speed of output. *S 1 will set minimum speed, *S 255 sets maximum. Just a bit of fun really. Sometimes it has it's uses in debugging or a fancy way of displaying a message.

***B** - BEEP command. *B 0 disables the beep and *B 1 re-enables it. *B on its own just beeps. Note also that CTRL-G or CHR$(7) will sound a beep as on most other micros.

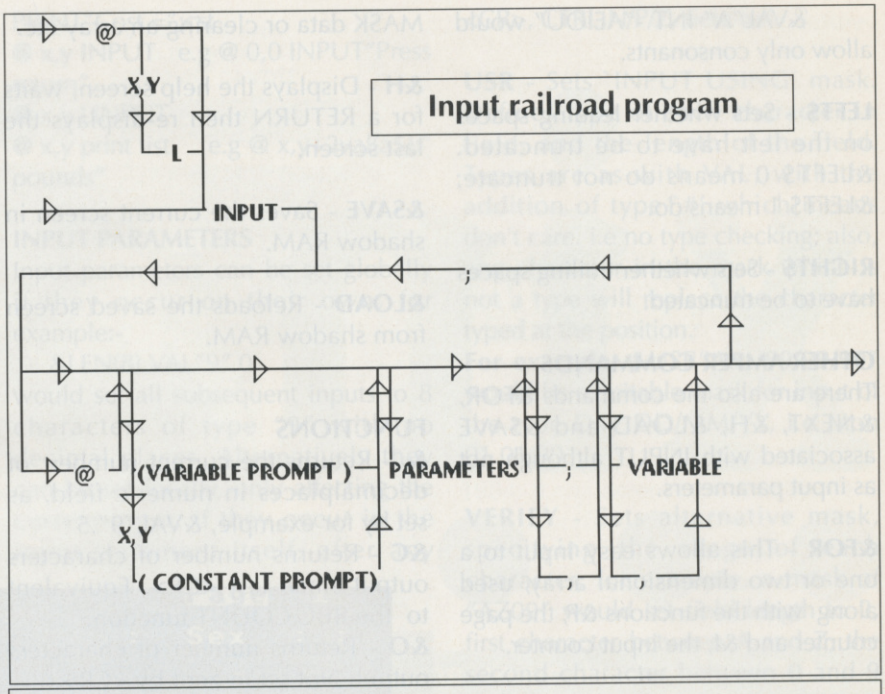
***R**- Equivalent to Apple INVERSE/NORMAL command. *R 1 sets reverse mode, not terminated by carriage return. *R 0 sets normal mode.

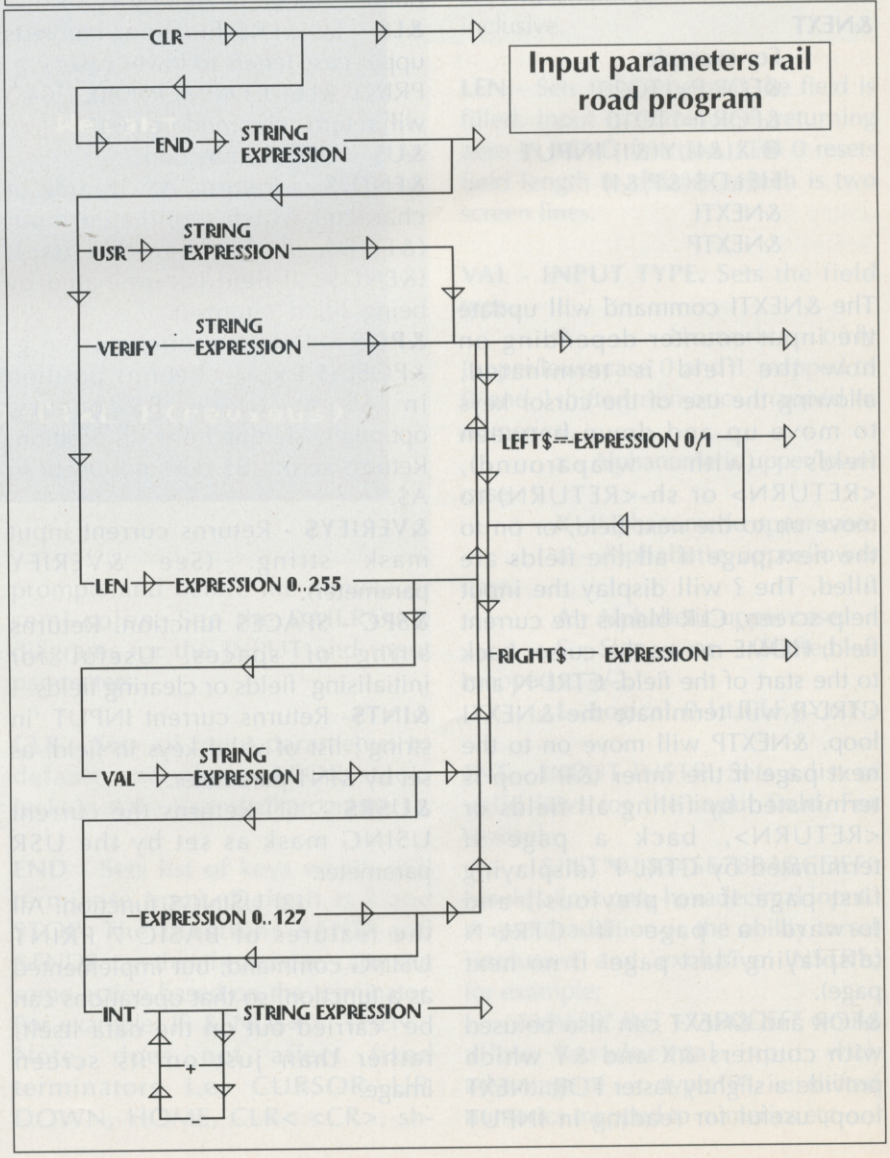
INPUT SYNTAX EXAMPLES

INPUT - Prints space and "?", waits for return.
INPUT,- Waits for return.
INPUT(A$),B$ - Displays prompt in A$, input into B$.
**INPUT LEN(8),CODE$** - Accepts 8 chars into CODE$.
**INPUT (P$(X)) VERIFY V$(X), I$(X)**- Display prompt in P$(X), accepts input using mask in V$(X) into I$(X).

See the input parameters railroad diagram for syntax of local parameters.

I hope I have not made all the above too difficult to understand. If you load the XINOUT program from outside the menu, you will be able to LIST it to see how things are put together.





Create your own variable joystick autofire by following this guide

JOYSTICK AUTOFIRE

YASIN POPTANI

There are many times when you put down your joystick and say, "I would have completed that if only the autofire was a touch slower." or you might say, "If only that autofire was faster and consistent."

Well what you are really asking for is an autofire which can be readily adjusted. There are joysticks within the market which have this facility but as with most facilitated joysticks they have excellent facilities but the joystick itself makes any game unplayable.

The simplicity of the Commodore's joyports allow you to create your own autofire but at the same time letting you use your favourite joystick.

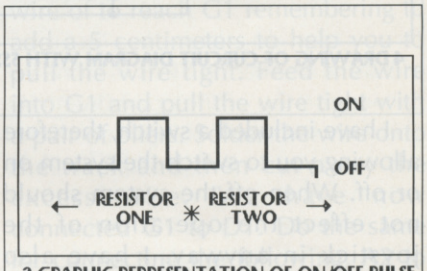
How! You all ask, well it's simple, all that has to be done is to put the autofire circuit in parallel to your joystick. Lets look as the Joyport.

The theory to the joystick is that as soon as a terminal, say left, contacts the Ground, a circuit is made and a left movement is made. This is done respectively with all the other terminals. This system is the basis of all joysticks.

The 555 Timer is a versatile timing chip, known as an astable device, this simply means that it can produce a consistent line of pulses, ideal for autofire. However, the 555 Timer chip has to be in the right format of circuit to do the job we want it to do. The proper configuration of circuit is as such;

By changing the value of the two resistors, which limits the amount of electricity going into the chip we can change the speed of the automatic fire. The top resistor controls the duration the fire is on and the bottom resistor controls the duration the fire is off. Just as below;

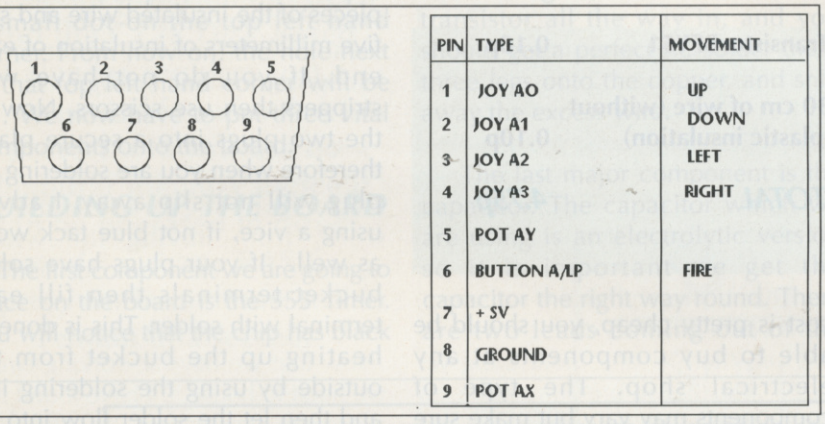
The capacitor is just a cell which stores electricity temporarily for the chip.



3 GRAPHIC REPRESENTATION OF ON/OFF PULSE WITH CAPTIONS SHOWING TWO RESISTORS.

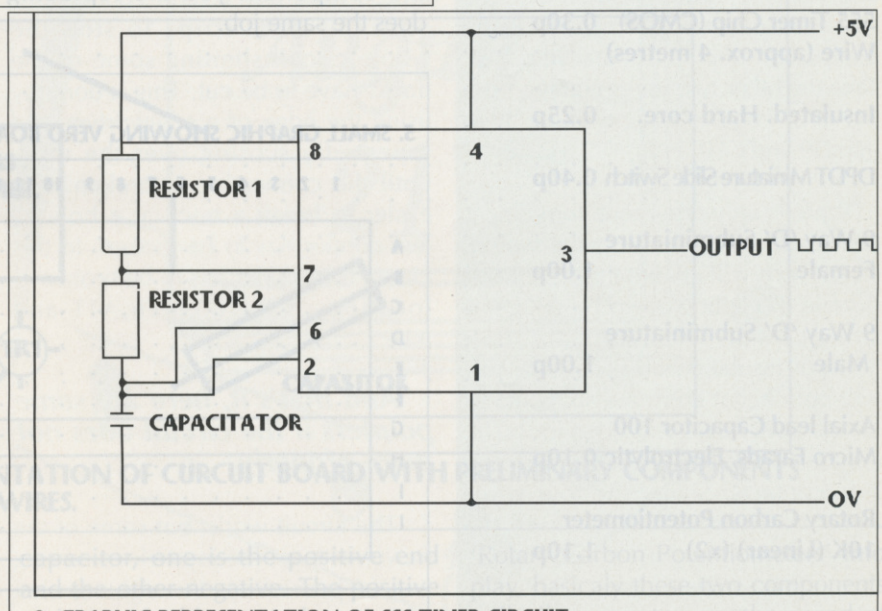
Now we know how we can produce the automatic fire we must incorporate this into a full circuit diagram.

Don't worry if you cannot understand the diagram it should become clearer when the circuit is constructed.

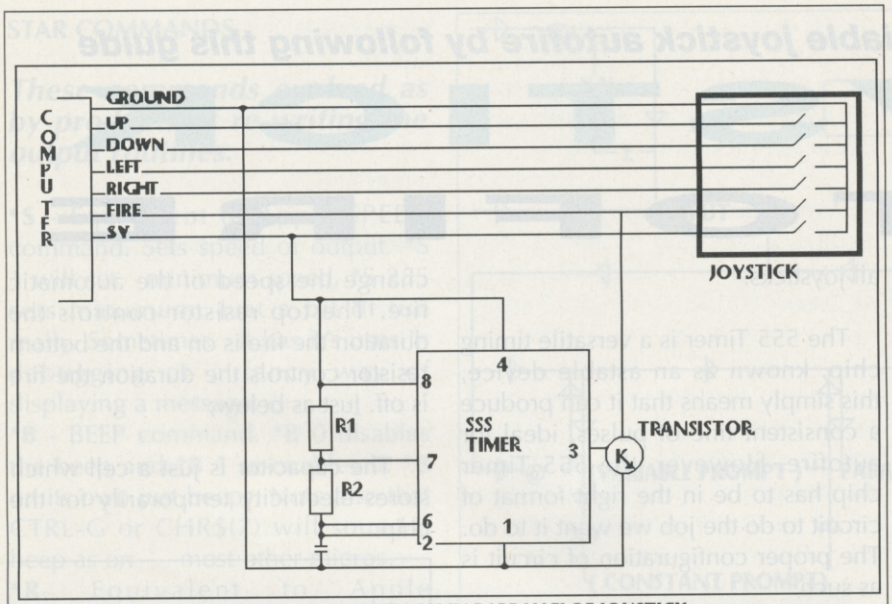


PIN	TYPE	MOVEMENT
1	JOY AO	UP
2	JOY A1	DOWN
3	JOY A2	LEFT
4	JOY A3	RIGHT
5	POT AY	
6	BUTTON A/LP	FIRE
7	+ 5V	
8	GROUND	
9	POT AX	

1. GRAPHIC REPRESENTATION OF JOYPORT AND NUMBERING OF PINS TO THIER DESIGNATED MOVEMENTS.



2. GRAPHIC REPRESENTATION OF 555 TIMER CIRCUIT.



4 DRAWING OF CURCUIT DIAGRAM WITH SSS IN PARRALLELOF JOYSTICK.

I have included a switch, therefore allowing you to switch the system on or off. When off the system should not effect the operation of the joystick in anyway. I have also included a Transistor, which acts as an electronic gate, switching the fire on and off.

Now we need to construct the system, here is a quick list of what is needed and how much it could cost.

As you can see the component cost is pretty cheap, you should be able to buy components at any electrical shop. The type of components may vary but make sure the component you are replacing does the same job.

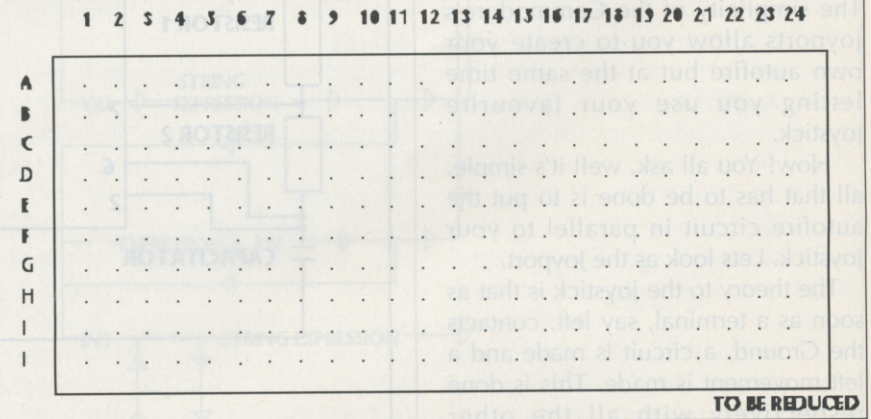
Components	*Price*
Veroboard (see below for size)	0.40p
555 Timer Chip (CMOS)	0.30p
Wire (approx. 4 metres)	
Insulated. Hard core.	0.25p
DPDT Miniature Slide Switch	0.40p
9 Way 'D' Subminiature Female	1.00p
9 Way 'D' Subminiature Male	1.00p
Axial lead Capacitor 100 Micro Farads, Electrolytic	0.10p
Rotary Carbon Potentiometer 10K (Linear) (x2)	1.10p
TransistorBFY51	0.10p
30 cm of wire (without plastic insulation)	0.10p
TOTAL	*4.75p*

The Veroboard should be a particular size otherwise you will not be able to fit all the components on. The size of the veroboard I am using is 3cm by 6cm or has 10 tracks with 24 holes on each track.

You will need access to a soldering iron and lots of solder. If you are new to soldering, you will first need to practice on a piece of scrap board, as the board we are going to use is extremely small and an overdose of solder on the board is likely to cause a short. The trick of soldering is to heat up the track first and then feed in the solder.

The first thing which has to be done is to solder up the two 9 pin plugs, which fit into the computers joyport. To do this first cut six 20cm, pieces of the insulated wire and strip five millimeters of insulation of each end. If you do not have wire strippers then use scissors. Now fix the two plugs into a secure place, therefore when you are soldering the plug will not slip away, I advise using a vice, if not blue tack works as well. If your plugs have solder bucket terminals then fill each terminal with solder. This is done by heating up the bucket from the outside by using the soldering iron and then let the solder flow into the bucket. This has to be done to both plugs. If you look carefully on the

5. SMALL GRAPHIC SHOWING VERO BOARD.



plug you should see numbers above the terminals, Solder the first wire into the terminal marked one. Then solder the other end of the wire into the other plugs terminal also marked one. Do this for terminals 2,3,4,5, and 9. Make sure you have got the right wire to the right terminal before proceeding.

Leave these two plugs in place for the time being, we shall come back to them.

The next step is to prepare the veroboard. However to avoid confusion it will be best to use co-ordinates. Here is a diagram of what the topside of the veroboard should look like. I have included co-ordinates which I shall use later.

Mark with a permanant black pen a small dot on the top left hand corner. From now on, the hole next to that top left hand corner will be A1. We now have to put three vital components onto the board.

BUILDING UP THE BOARD

The first component we are going to place on the board is the 555 Timer. You will notice that the chip has black dot or indentation. Turn the chip around so that this black dot/indentation is facing upwards. Now place the chip onto the board so the top left pin is in C5 and the bottom right pin is in G8. Hold the chip firmly in place and turn the board over, on this side you should see copper tracks running along the board. Solder the chip into place, do this neatly by not allowing to much solder onto the board. Solder all of the eight terminals.

The next component is the Transistor. As you can see the transistor has three legs, the collector, the emitter and the base. The base and the emitter have been insulated by a coloured piece of plastic. Next to the emitter is a small metal flag. The collector is has no plastic insulation. By using these pointers find which leg is which. The base leg goes in at I8, the emitter goes in at J7 and the collector goes in at H7. Push the transistor all the way in, and you should get a perfect fit, solder these three legs onto the copper, and snip away the excess wire.

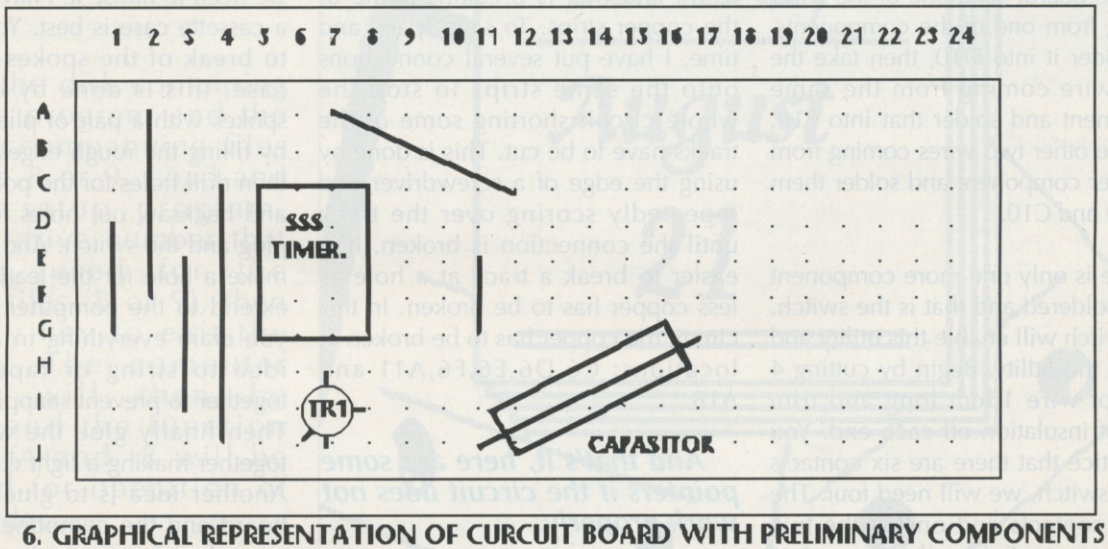
The last major component is the capacitor. The capacitor which we are using is an electrolytic version so it is important we get the capacitor the right way round. There are two leads coming out of the capacitor, one is the positive end and the other negative. The positive end is marked by an indentation on the capacitor, this end of the capacitor goes in G17. The other end, the negative end goes in J11. Solder the two leads onto the copper and then snip away the excess wire.

SOLDER AT THE READY

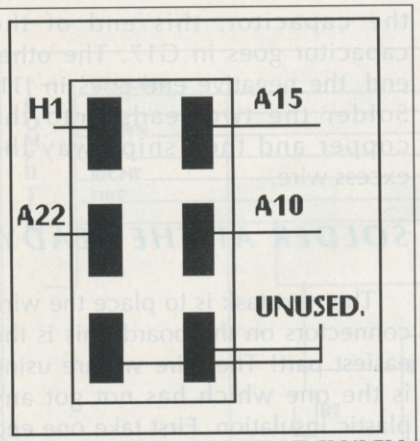
The next task is to place the wire connectors on the board, this is the easiest part! The wire we are using is the one which has not got any plastic insulation. First take one end of the wire and feed in 10mm of wire through D1. Solder the wire into place, and then cut enough wire of to reach G1 remembering to add a 5 centimeters to help you to pull the wire tight. Feed the wire into G1 and pull the wire tight with a pair of pliers. Solder the wire onto the track and then cut away the excess wire. We have now connected G1 to D1. Do the same for connecting the following; A2 to F2, E3 to I3, C4 to J4, E11 to G11 and C12 to A7.

Your circuit board should now look like Diagram 6 at the bottom of the page.

The next task is to put the two 'Rotary Carbon Potentiometers' into play, basicaly these two components will allow you to vary the speed of



6. GRAPHICAL REPRESENTATION OF CURCUIT BOARD WITH PRELIMINARY COMPONENTS AND CONNECTION WIRES.



7. GRAPHIC SHOWING SWICH AND TERMINALS WITH END LOCATIONS.

fire, they do this by changing electrical resistance. This is done by turning the top of the component. As you can see these components have three connectors, however we shall only need two, the centre connector pin and a side connector. It does not matter which side connector you use, but make sure that you select the same connector for each component.

Cut four strips of wire each 15cm long and trim 5mm of insulation off each end. Then solder the strips onto the components, leaving you with four wires. These wires have to be soldered onto the board. Take one of the wires coming from one of the components, and solder it into D10, then take the other wire coming from the same component and solder that into E10. Take the other two wires coming from the other component and solder them into D9 and C10.

There is only one more component left unsoldered and that is the switch. This switch will enable this utility and disable the utility. Begin by cutting 4 strips of wire 15cm long and trim 5mm of insulation off each end. You will notice that there are six contacts on the switch, we will need four. The centre contacts and one of the two side contacts. Solder the four wires onto the switch. Take each of the wires from the switch and solder them as shown next.

Now we have to do the final connections which will make the thing work. Remember the 9 pin 'D' plugs, we are now going to connect them on to the board. First establish which plug is male and which is Female. The female plug is the one which plugs into the computer. Cut three strips of insulated wire measuring 20cm and three strips of wire measuring 15cm. Strip 5mm of insulation of each end. Take a strip measuring 20cm and solder it onto pin no 6 in the female plug. Take the other end and solder that on the board at location A24. Take a strip measuring 15cm and solder it onto pin no 6 in the male plug. Solder the other end on the board at location A19. Take a strip measuring 20cm and solder it into pin 7 of the female plug. Then solder the other end into location A17. Solder a 15cm wire onto the male plug at pin no. 7 and solder the other end into locaation A13. Take the last 20cm strip and solder that into pin no 8 of the female plug, and solder the other end into location J24. The last wire goes into the male plug at pin no 8 and the other end goes into location J1.

That is the end of the soldering, however do not try the circuit out just yet, there is one more thing yet to do and that is breaking some of the copper strips. To save space and time, I have put several connections onto the same strip. To stop the whole circuit shorting some of the tracks have to be cut. This is done by using the edge of a screwdriver and repeatedly scoring over the track until the connection is broken. It is easier to break a track at a hole as less copper has to be broken. In this circuit the copper has to be broken at locations; C6,D6,E6,F6,A11 and A18.

And that's it, here are some pointers if the circuit does not work properly.

- The circuit works but some of the movements on the joystick are wrong say if I pushed up, the sights go left and so on.

* Some of the connections connecting the two D Plugs have been mixed up, the best thing here is to desolder the above five wires and redo them.

- Nothing works, even the joystick is dead.

* The ground has been broken, check the wires at terminals J1 and J24 and see if there is any continuity.

- It just does not work, however the joystick is fine.

* There could be several problems here, the first could be that one of the connections could have broken, as the wire used, solid core wire, is good, however has a tendency to snap. The next reason, could be that the components or wires could have been inserted wrongly, check them. The last thing, could be that the Potentiometer is set at zero. That means that the component is not letting any current through, turn the dial and the circuit should work.

There could be umpteen more reasons why the circuit could not work, from having the wrong components to having dodgy connections.

Now that we have a working circuit we need to house it. I have found that a cassette case is best. You first have to break of the spokes within the case, this is done by cutting the spokes with a pair of pliers and then by filling the rough edges down. You then drill holes for the potentiometers and hacksaw out holes for the male plug and the switch. And then finally make a hole for the lead which will extend to the computer. But before you cram everything in it is a good idea to string or tape the wire together to prevent snapping of wires. Then finally glue the whole thing together making a tight compact unit. Another idea is to glue the circuit board and the components into the case therefore stopping the whole thing rattling around, but make sure that the glue that you are using cannot conduct electricity, otherwise you might have a short.

C128 CALENDAR

Why pay out good money for a calendar when you can make your own for nothing. This issue Paul Traynor shows you how !

This program uses a total of 13 files, 1 for each month and a setup program. Once the setup program has been run the C128 Disk-Based Calendar requires just a couple of function key presses and will use no computer memory hence it can be operated in direct mode and any BASIC program in memory will remain unaffected.

Setup

First put the disk containing the setup program and the twelve accompanying files into drive 8 or 9. Load and run the setup program, enter the drive number that is being used for the calendar files (8 or 9), then press any key to exit the program. The computer memory will then be cleared and the function keys f1, f4 and f8 will be redefined for operation of the calendar.

Operation

To operate the calendar;
1. press F1, enter the first 3 letters of the desired month, then return.
2. press F8, the file will execute and the desired month will be displayed.
3. press F4, the screen is cleared and control is back with the user.

It is possible to operate the calendar from within your own programs by using BASIC program lines which simulate the new key definitions, you will have to have a string variable set by the user for the month title, which can then go into the open statement.

GOMOKU

Beat the computer in this well known game of stratergy.

ADRIAN MILLETT

I can hear a few groans from the older readers of this mag, as I type this. "Wot, another line-em-up strategy game?" Well, it's true, it is another redo of a traditional strategy game. However, I hope you will find that some of the new ideas I have put into this one, redeems it somewhat. The original version of GO-MOKU is played on the same 19 by 19 board that the fascinating (and very complex) chinese game of "Go" is played on, and simply involves two players (one black, one white) placing pieces on a board until one player wins by getting 5-in-a-line in any vertical, horizontal or diagonal direction (a bit like a giant noughts and crosses). When you first run the game you are presented with a moderately interesting title screen, and are invited to hit the SPACE bar. You are then given a command summary, and you can start a game by selecting "T" for traditional GO-MOKU or "H" for a "Hard" variation of the game I have devised. (See "Game-play theory" below.) I recommend you try the "Hard" version - It isn't really any harder to play, just more interesting.

I don't want to ramble on too much in a "this is the key you press" command list, since if you have enough wit to switch on the computer and load the disk, the commands won't pose a problem. Remember that if you do have problems, the editor runs a special 3am help-line from the comfort of his own bidet. (What do you mean I'm fired?)

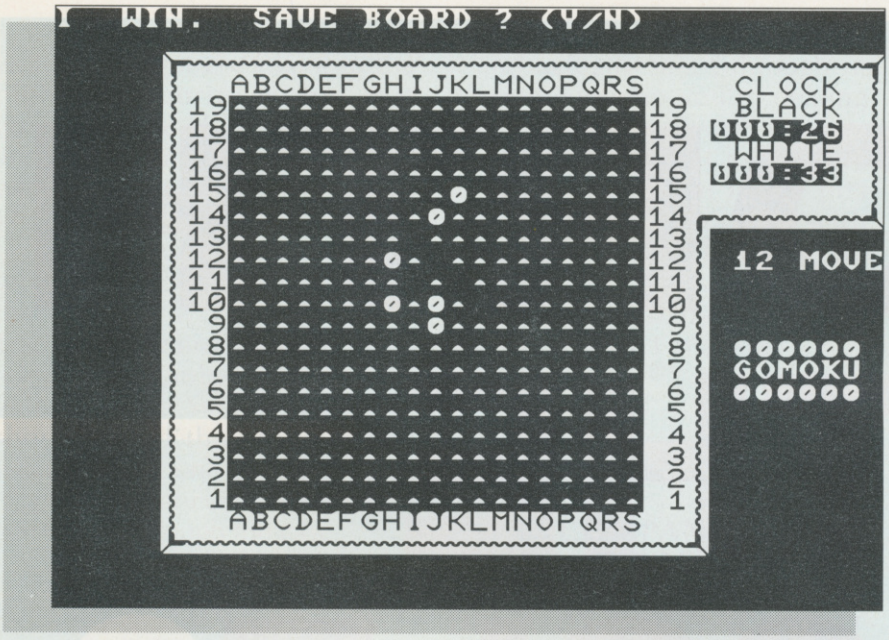
```
        Command Summary

  Gomoku is a simple and yet intriguing
strategy game. The 1st player to get
5 pieces in a line wins the game.
Use the cursor keys to move cursor, and
hit RETURN to place your piece.

F1,F3,F5   - Change screen, border, ink.
F2,F4      - Select piece colours.
SHIFT-R    - Restart a new game.
SHIFT-X    - Exchange colours.
SHIFT-G    - Make computer take a move
SHIFT-A    - Automatic computer play,
             Hit SPACE-BAR to end
SHIFT-E,W,B - Erase/Place piece at crsr
SHIFT-L,S  - Load/Save board to memory
1,2        - Select 1 or 2 human players
SELECT 'T' (Trad) OR 'H' (Hard) GAME.
```

THE COMMANDS ARE AS FOLLOWS

Cursor-keys - Select a square,
Return - Make a move at the selected square.
F1,F3,F5 - Screen,border or ink colours.
F2,F4 - Select piece colours.
Shift-R - Restart a new game.
Shift-X - Xchange player colour. (White <-> Black)
Shift-G - Make computer take your turn.
Shift-A - Automatic computer play against itself, hit SPACE-BAR to end.
Shift-E - Erase piece at cursor position.
Shift-W - Place a white piece at cursor position.
Shift-B - Place a black piece at cursor position.
Shift-S - Save board to mem.
Shift-L - Load board from mem .
2 - 2 human players.
1 - 1 human player vs computer (normal)

GAME-PLAY THEORY

Since a line of five wins, if you get a row of four that is unblocked at each end, you have effectively won the game, unless your opponent can win immediately by making a row of five. This is because whatever end your opponent blocks, you can simply move to the other end and win. By similar reasoning, 2 open-ended rows of three pieces usually poses a winning threat, because your opponent cannot stop you making an open-ended row of four out of one of these rows of three. Now, after serious analysis, players of traditional GO-MOKU found that whoever moves first should be able to force a win early in the game by making a double-threat of three pieces. For this reason, the Japanese have invented a variation of the game called Renju. In this game the first player to move (White) is restricted in that he



cannot make moves that form certain combinations of multiple threats of three or more pieces, and the board itself is limited to 15 by 15 squares. Now a certain amount of reflection on this game has lead me to believe that it is really a bit of a botch. For a start, you have restrictions which apply to only one player, which in itself is unnatural. Further, the restriction of simply depriving a side of his strongest moves surely must put that side into a somewhat passive role. What I felt was needed was a natural extension that doesn't interfere with the game's basic concept. I have tried to achieve this by changing the board itself, and making certain squares "illegal", ie. neither side can play on them. If you run the game and select "H" for the hard version you will be able to see the pattern I have decided to use. Essentially I have placed an illegal square at the intersection of every fourth column and row, counting from the 2nd row and column from the top. What I hope this achieves is a "breaking up" of the boards continuity, making it impossible (?) for the white player to force a win every time. This pattern also has the effect of making some squares stronger than others, and I have put suitable weights into the computers game-play. However I will leave you the fun of discovering the relative strength and weakness of these areas for yourself!

CDU
BACK ISSUES

Back numbers of Commodore Disk User are avaliable from:-

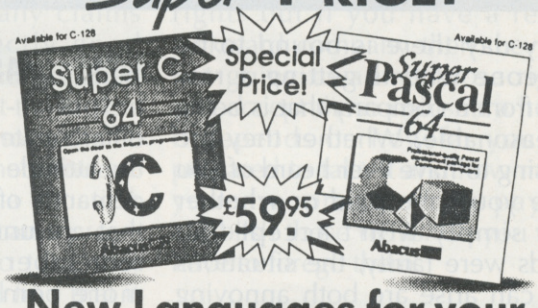
SELECT SUBSCRIPTIONS LTD
5 RIVER PARK ESTATE
BERKHAMSTED
HERTS
HP4 1HL

TEL (0442) 876661

Price:- £3.25 INC Post + Packaging

CHEQUES PAYABLE TO ALPHAVITE PUBLICATIONS LTD

Super COMBO Super PRICE!



New Languages for your C-64 and C-128

Super Pascal-64	£39.95	Super Pascal-128	£39.95
Super C-64	£39.95	Super C-128	£39.95
Cobol 64/128	£29.95	Fortran-64	£29.95
ADA-64	£19.95	Forth-64	£12.95
LOGO-64	£14.95	Pilot-64	£14.95
Video Basic-64	£19.95	Pascal-64	£14.95
Zoom Pascal-	£9.95	Assembler/Monitor-64	£9.95

SPECIAL OFFER Super C + Super Pascal (C64 or C128) £59.95

Send S.A.E. for full details

ADAMSOFT, 18 NORWICH AVENUE, ROCHDALE, LANCS OL11 5JZ
Access/Visa orders Tel: 0706 524304

FEATURE

WORTH THE HASSLE?

Bad service, shoddy goods, poor after sales service. We all experience some or all of these at some time. What can we do about it? Our investigation brings to light some worrying findings.

JASON FINCH

Every day there is bound to be someone who is getting a raw deal from a company that is being unreasonable. Whether they are refusing to have ever heard of you once you have paid or whether they simply won't accept that goods were faulty, the situations that can arise are both annoying and frustrating. I would like to thank Mr.Jonathan Lines of Hodson and Lines Solicitors in Rugby for allowing me to interview him and also all the companies that have offered me the experience and service that became an integral part of the basis for some of the questions.

Anyway, this is a two part article, the second part being featured next month. There follows the above mentioned interview, the text in bold being my part of the dialogue. I hope that you find it both interesting and perhaps helpful - it may help to differentiate between whether it is the attitude of the company or the demands of the customer himself that are unreasonable. Please do remember that there are a lot more companies that provide excellent service than we may think. If you do come across an awkward bunch of employees then you must decide yourself to what levels you will climb and what is and what is not worth the hassle that is so likely to be involved. So let the talks begin...

EDITORS COMMENT

Before we carry on, may I just say that there are some companies that bend over backwards to help their customers. One such company is BELSTAFF INTERNATIONAL, manufacturers of motorcycle clothing. Without going into details, I would like to publicly thank BELSTAFF, and Mrs. O. GAMBLE in particular, for their prompt, courteous and effecient manner in which they deal with their customers.....

COMPLAINING

MANY COMPANIES FORGET THE IMPORTANCE OF COMMON COURTESY ONCE THEY HAVE THE CUSTOMER'S MONEY IN THEIR POCKETS. IS THERE ANYWHERE TO WHICH OFFICIAL COMPLAINTS ABOUT SERVICE CAN BE SENT AND IF SO WHAT IS THE USUAL PROCEDURE?

If the customer is getting absolutely nowhere from talking to the company then he should

write a letter of complaint to the highest authority within that company. He should be polite and explain briefly but completely what the problem is. He should keep a copy of it and send it recorded delivery. If no satisfaction is obtained thereafter then he can write to his local Trading Standards Office. Their telephone number and address will be in the Yellow Pages. He should detail the problem and what he has done and then there is a good chance that the office can follow through the case.

MONEY BACK

IF THE CUSTOMER IS RECEIVING A GREAT DEAL OF HASSLE FROM A COMPANY AND DECIDES THAT HE WOULD LIKE HIS MONEY BACK SO THAT HE CAN TRY ELSEWHERE IN THE PURCHASING OF HIS DESIRED GOODS, IS HE ENTITLED TO ASK THE COMPANY FOR A FULL REFUND OF HIS MONEY?

If the company is posing as more than just a minor irritation, say, then I would have thought that that could justify you writing and demanding your money back. It's another matter whether or not the company agrees to that or how easy or difficult it is to enforce that right - that's another story altogether. You may have a right but if the company doesn't agree to it then you may, if you're not prepared to, as a last resort, take it to court and pursue that right with all the extra costs that go with that and all the rest of it, you may find that you give in without actually getting that right. That's the nature of the circumstances. It's one thing to have the right. It's another thing to be in a position to enforce it and actually see it through. It depends on who you're dealing with and how awkward they try and choose to be. I mean here we're assuming that in the first place we are dealing with someone who is being a bit uncooperative, and so the right is one thing - actually getting the result is another, I'm afraid.

IT DOESN'T WORK

IF A COMPANY SENDS FAULTY GOODS, IS THE CUSTOMER ENTITLED TO DEMAND A REPLACEMENT OR A FREE REPAIR, EVEN THOUGH HE MAY BE UNABLE TO PROVE THAT IT WAS FAULTY WHEN HE RECEIVED IT, SAY IF HE RETURNED IT WITHIN A WEEK?

In theory, yes. I say "in theory" because if the company claims that it was in perfectly good condition, it wasn't faulty, when they sent it out and that the fault is not their fault, he may have difficulty in proving the contrary. You have just said in a situation where the customer can't prove that. If they choose to take that point then he may well find that it is very difficult to get the company to accept the liability. That is the problem inherent to the one. But in theory, if it is in fact the company's fault then you ought to have the right to get them to take it back or replace it depending upon the circumstances. But if you can't prove your case then you can't necessarily enforce your right. If you are in this sort of situation then there are two other factors, I think. One is who actually has to do the proving in the first place. If you are trying to prove that the company is at fault then it is on your head, because you are making the claim, and it is your responsibility to prove that they are at fault, then if you can't do that, they are never called upon to prove that it wasn't them. On the other hand, even if you can't prove that, if the circumstances are such that it would appear to be the company and in the last resort if it came to a court they would assume that it was the company and then the whole situation is turned around and it would be for the company to prove that it wasn't them because everything would seem to indicate that it was, and then they have got the problem. But in the first instance you've got the problem of proving that it was them. Who has to actually prove the thing in the first place is an important factor there. How readily either side does or doesn't accept the situation, how far you are prepared to take it, all balances out. I keep on saying that you have rights here and there, and there are some various very good rights but if you have a really awkward company then they are not going to admit those rights and accept those rights in a hurry. It depends on how far you are prepared to take it and how much you are prepared to pay to establish those rights. Or you may have to forego them.

WHO ARE YOU?

SAY FOR INSTANCE THAT A COMPANY CHARGES YOUR CREDIT CARD OR CASHES YOUR CHEQUE AND THEN EITHER REFUSES TO SEND THE GOODS OR CLAIMS THAT IT HAS NEVER HEARD FROM YOU, IS THERE ANYTHING THAT THE CUSTOMER CAN DO? ARE THERE ANY LAWS THAT CAN BECOME INVOLVED?

Well, that is fairly blatant - a situation like that. And, yes, there we have a simple Law of Contracts. The customer has a contract for the company to supply the goods and then customer to pay for them. The customer has paid and the company hasn't supplied the goods. So you sue them under the Law of Contracts - it is as simple as that.

BUT IF THE CUSTOMER HAS NO WAY TO PROVE THE CONTRARY - WHAT THEN?

In that sort of situation it comes down to who the courts would believe, if it came to court - which in a situation like that it is quite likely to because if that happened to me, if there was more than about ten or twenty pounds involved, and there is likely to be hundreds of pounds involved in these sort of situations then if I had paid that sort of money for a piece of hardware and I didn't get it, I am not going to simply accept the company's answer saying "Well we did send it to you". I would be sueing them in court no trouble at all and I would have every confidence and hope that the court would believe my story even if - well, how do you prove a negative?

IT'S GOING BACK

IF A COMPANY SENDS YOU SOME GOODS AND THEY ARRIVE BY COURIER, AND UPON TESTING YOU FIND THAT THEY ARE FAULTY, ARE YOU ENTITLED TO DEMAND THAT THE SAME COURIER RETURNS TO COLLECT THEM OR CLAIM A REFUND OF THE POSTAGE, BEARING IN MIND THAT IT WAS THE COMPANY AT FAULT?

That could depend on the terms of the particular contract of the company and how much they value the customer's trade. If there are terms in writing, and most companies have written terms in dealing with the public, and unless the term is particularly unreasonable, the chances are that that sort of thing would be answered in that. If there is no such term, I am not sure that there is any overriding principle that says that the company has got to pay that sort of thing. Obviously it would be courteous if they did, seeing as how the customer is not to blame for having to return the goods. If you don't inquire first, the thing to do would be to claim it and then if they do everything else right bar that then, in practice, you forget it. That is a matter of practicality.

BUT THE POST OFFICE'S PARCEL SERVICE IS SLOW, ALTHOUGH CHEAP, AND A COURIER COULD COST UP TO THIRTY OR EVEN FORTY POUNDS.

Well, yes ok.

AND IF IT IS FAULTY WHEN YOU RECEIVE IT, THEN IT'S NOT NECESSARILY YOUR FAULT IS IT?

You include that as part of your claim if in fact you are returning it by the same means, but as I say, in the end if they refuse, whether you end up sueing them for the cost of the courier to return it to them - yes, I think, you could have a reasonable case. But at the end of the day it is a matter of judgement. There is however the fact that with mail order catalogues when you pay after having received and checked the goods, you only need to inform the company that they are faulty, you don't want them and that they are available to be collected. It's then up to the company to collect the goods. You must check in those circumstances that the catalogue does not state otherwise though.

CHANGING YOUR MIND

IF, HAVING RECEIVED THE GOODS, THE CUSTOMER DECIDES IT IS NOT REALLY WHAT HE WANTED, IS HE ENTITLED TO SAY "WELL THIS IS NOT WHAT I EXPECTED IT TO BE" AND THEREFORE GET A REFUND OR MUST HE JUST ACCEPT IT AND THAT BE THE END OF IT?

There are two aspects to this, aren't there. If the goods are substantially different from what the company represented them to be - if the company said they would be able to perform a certain function and they don't - then he is clearly entitled to a refund. If on the other hand it is his misjudgement and he buys them thinking, from what he understands from what the company said about them, that they are going to do the job he wants, but when he gets the goods home he realises that he has analysed it wrong, then it is a different story. Now you may get plenty of people that are not prepared to admit that they have made a mistake in that case and they blame it on the company although it's their fault really. In that situation they are obviously not entitled to it. It is all a question of who has made the misrepresentation or misjudgement. So that's the answer there.

TO BE CONTINUED NEXT MONTH....

SMOOTH SCROLL DEMO

A simple demonstration showing the techniques used for GOMOKU loading screen

ADRIAN MILLETT

From time to time letters appear in journals like CDU asking how particular effects in programs are acheived. Since I have recently written a general purpose title screen to jazz up some of my programs, I felt it would be a good idea to publish it and describe how to design these sort of effects in more detail. I hasten to add that there is nothing revolutionary about this smooth-scroller. However the pleasing video effects of a "bouncing" smooth scroll set against a moving background are achieved with a few straight-forward machine code routines.

DOWN TO BUSINESS

When you run the program "DEMO SMOOTH", which is a front-end basic loader, it will load the following files:-

The machine code for the video effects "SMOOTHxx.EXE" (Loc $9C00-9FFF).

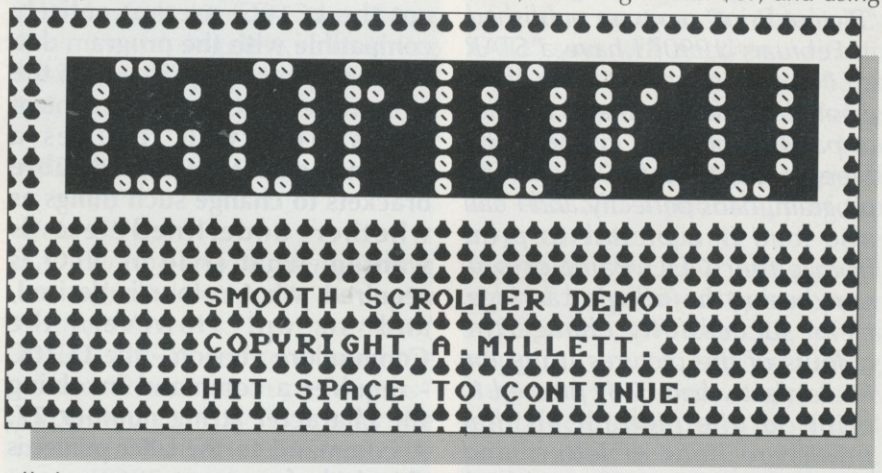
The user defined character set "SET.GOM.xx" ($8800-$8FFF).

A special sprite-sized character set for the smooth scroll called "LET.$A000" (A000-AC00).

Incidentally the "xx" in these file names is a version number. Examination of the disk directory will reveal the final versions used. The program sets basic memory-top to $8000, which becomes the new address for screen and VIC memory. After calling routines to initialise various variables and arrays (See lines 300-380) the main scroll-set up routine at 20000 is executed. You can modify this to design title screens for your own programs!

First, the screen is filled with the "moving background" character, screen code 87 in black ink, by calling a FILLVDU routine at 29000. After that, a suitable title screen is printed over it using normal BASIC print statements. Next, a message is set up for the smooth scroller, (20310-20390) by repeatedly loading MSG$ and calling GOSUB 32700. Note that embedded control characters can be used to toggle FAST/SLOW scroll (%) and BOUNCY scroll (&). Finally the main SCROLL.A.MESSAGE routine at 32100 is called, which executes the scroll m/code until one of the characters specified in KY$ is hit.

If you examine the listing printed here, you will find out how to access all the principle m/code routines in the jump-vector table ($9C00-$9C30), and also how to change the codes flags and parameters. For instance the character used for the moving background can be altered by changing the contents of the ROTCHAR variable at $9C43, ie. POKE 40003,90. Up to now I have described the methods of access from the point of view of BASIC, however the same principles apply if you are calling these routines from another m/code program. Indeed, if anything, many things are simplified. For instance, you can simply set the MSGLOC pointer (at $9C32) to point to where your null-terminated smooth-scroll message is located in memory, without having to move anything around.



THE REST IS UP TO YOU

Many of the neatest video effects you see in programs and demos use quite simple methods. For instance, the "moving background" effect here is done by filling the background with a character, and then rotating the bytes of that character vertically. The "bouncy" scroll is achieved by taking the horizontal position of a sprite letter, AND'ing it with $0F, and using this value as an index to the look-up table called WIGGLE. This value can then be added to the vertical position of the sprite. Extensions of this technique could be used to obtain quite exotic sprite movements.

It is important to synchronise these effects to the screen refresh to achieve smoothness of motion. Often this is done with a raster interrupt. In this case, for reasons of clarity, I decided to use a simpler routine (WAIT1SCR) that scans the raster register until a particular value comes around.

OK you've got a title screen, so all you need to do is design the rest of the program..

TECHNO-INFO

If you have a problem, If no-one can help.. Then maybe you can hire the T-Team.

PICTURE PRINT

Dear CDU,

Could you please tell me whether or not there is a bug in the "Picture Print" program published in February 1990? I have a STAR LC10 colour printer with a parallel interface and also a serial to parallel interface converter by Meedmore, type 92008/G. The program loads perfectly and I can also load in a picture to print from either CDU Paint or one saved using the Trilogic cartridge. I can successfully change the colours of the pictures using the F1 key but when F7 is pressed to print out the picture, all that prints are lines of letters and graphic symbols. I do not think that the printer is at fault as it successfully prints pictures directly from CDU Paint and from the picture dump on the Trilogic cartridge. I look forward to hearing from you with a possible solution to the above problem.

David Paddison, Stoke Poges.

JASON FINCH

Dear David,

There are no bugs in the program and this has been backed up by the readers that have sent us samples of pictures produced by the program. It would seem from experience that the standard colour models (ie: not the LC10C) are not entirely compatible with the program due to the different ways in which the two printers receive their control codes. The LC10 requires a command involving double brackets to change such things as whether underlined text is wanted, what style of NLQ is required, what colour is desired, and so on. However, the Commodore version - the LC10C - requires a command involving the character string number 27. A command to the latter printer is given in the form PRINT#4,CHR$(27)CHR$(114)CHR$(1) and not as letters within double brackets. Therefore the control codes that are sent control an LC10C and put it in the correct mode of operation for printing out a bitmap screen will not have the same effect on an LC10 and that is why no picture is printed. The reason that it would work with the cartridge is because that is likely to send out a very simple command that is recognised by all printers to put them in dot graphics mode. Unfortunately, unless the program is changed vastly, I cannot see any possible solution - unless you wish to sell your printer and buy a CBM standard version, which I doubt will be the case. Sorry that I cannot be of any more help

SUBMISSIONS

Dear CDU,

I have written programs for both the Commodore 128 and Commodore 64. I would like to have them published in CDU. But a problem occurred - since you have changed address I do not know where to send them or what procedures to take. Are they still the same as printed on page 30 of April's CDU? I would be grateful if you helped me out.

Mosthak Ahmed, Milton Keynes.

Dear Ahmed,

The procedures are indeed exactly the same as before and the address to which you should send them is as follows: CDU(Submissions), Alphavite Publications Ltd., 20 Potters Lane, Kiln Farm, Milton Keynes, MK11 3HF. Good luck in getting them published. (See the article CONTRIBUTIONS in this months issue....Ed!!!)

LETTERS

LOADING PROBLEMS

Dear CDU,
I have recently bought an Oceanic disk drive for use with my C64c and STAR LC10 parallel printer which is connected to my 64 via a RAM centronics interface. The problem occurs when I try to load certain disks. As the disk drive is new, I have only a few disks. Both Mini Office II and Superbase work well and so do all the appropriate functions (loading, saving files, etc). The games disks I have, however, do not work. I have 100% Dynamite by Ocean and IK+ by System 3. With these, when I type in the load command LOAD"",8,1 the disk drive activates and after a few seconds I get the File Not Found error. I suspect this may be the same problem as two letters published a short while ago where you suggested a disk editor as the remedy. If this is so, could you please tell me where I can obtain one from. Please try to help. Thanks for any assistance.*
Mr. C.Lowe, Cleveland.

Dear Mr.Lowe,
Firstly, the problem is not the same as those two that you mentioned. The error resulted in those cases because of the searching procedure of the 1571 disk drive and because the files were protected (less than sign after the filename). There are a number of things you could try in order to help pin down the problem. Firstly, if you have any sort of backup cartridge plugged into the computer, take it out and try again. If that makes no difference or you do not have one, try switching off any special DOS within your disk drive. That also applies only if you have had one specially fitted (such as Dolphin DOS). If none of that either applies or works, then unplug everything from your system other than the monitor and disk drive, and link the drive directly to your computer. That should tell you whether or not having the printer in the line makes any difference - don't just switch the printer off, disconnect it completely. If having the computer linked directly to the drive does not work then the problem is either the disk or the drive. If you can, try the disks on someone elses system (someone with a different drive). If that is not possible then let us assume that it is the drive at fault. All you can do is take it back to where you bought it with your disks and show them what happens. This sort of thing happened with my first drive - an Excellerator, the same thing as the Oceanic. They should be able to give you a replacement. If that still doesn't work then perhaps they could suggest something having seen what happens with the second drive.

MOUSE PROBLEMS

Dear CDU,
I am writing to you asking if you can help me! I have a Commodore 1351 proportional mouse and I need to control the pointer from "Interrupt Pointers" published in May 1990. As the mouse is not involved with the same POKE number as a joystick, I wondered if you could make a little program to help me along. Also I have tried everywhere to get a copy of the program published many years ago in CDU called "C-CAD" that enabled you to draw circuit diagrams. I have bought the magazine and unfortunately formatted the software. Now don't bother saying it is possible to get a copy from Protoscan as they don't wish to know. I have bought your magazine from the very first issue and am really pleased to hear that isn't going to be stopped after all. However, I do have a complaint about the magazine. In nearly all the issues you publish yet another BASIC extension... Can we have something more original please!
Stephen Bagnall, Stafford.

Dear Stephen,
Unfortunately there is no simple change you can make to ensure that the pointer can be controlled by the mouse. What is needed is for the demo on the disk supplied with the mouse to be altered and then the interrupts "integrated". Doing this easily is not possible because you also need to incorporate your own checks for boundaries. To do what you want would involve me completely altering the code and republishing the entire program. Sorry that I can't provide practical help in the way of a new routine. To obtain "C-CAD" you will need to write to Alphavite and ask about back issues beyond those available from Select Subscriptions. You should be able to get a copy of the disk and a photocopy of the appropriate parts of the magazine. On your last point I would be pleased to see your evidence. Of the 22 disks supplied up to and including August 1990, only eight have contained some form of what I would class a BASIC extension. And you may be surprised by the actual statistics. Since the first issue and up to the August disk again, nearly 14600 blocks of disk space have been used by programs and that does not include the menu systems, simple file copiers and 'read me' files - and only 222 of those have been part of a BASIC extension. Strictly speaking there have been 24 disks because two were double sided. This is just one and a half percent of the program space used by them. I am sorry, but I don't share your opinion on extended BASICs. Also, we can only publish what we receive

LETTERS

from the readers. I hope that you have some luck with the mouse driver program.

COLOUR PRINTERS

Dear CDU,
I do not have a great deal of knowledge of printers, but I have heard about a STAR LC10C printer which prints in colour. Firstly, is this true, and secondly, would it be able to print images created with the CDU Paint program, featured in the March/April 1989 issue? If this printer does not print in colour, could you let me know of any printers (dot matrix) which do print in colour, and will also print images from CDU Paint. This would be of invaluable help to me as I require something like this for college work for printing out graphs, pie-charts, etc, which ideally require colour. Thanking you in anticipation.
Mike Pitches, Plymouth.

Dear Mike,
There is indeed a STAR LC10C which prints in colour and with the correct program it will dump a CDU Paint file in colour. The necessary program for that is "Picture Print" published in the February 1990 issue, or alternatively you could purchase the Super Snapshot cartridge from FSSL. The telephone number is 0386-553153. That would simply involve you pressing a button and selecting a few options once the picture that you wanted was displayed on the screen. You must ensure that when purchasing the printer you quote the "C" on the end of "LC10C" or you will experience problems with printing whole screens in colour.

NETWORKING

Dear CDU,
I have been reading your magazine since the beginning of this year and I think it is great. I have two 64s, an Oceanic drive and a Citizen 120D printer. I am writing because I have no idea how to set up a computer network with my two 64s (via some sort of cable). So could you please tell me how to program the computer (preferably in BASIC, if not in machine code) to obtain the results and anything else that I may need to know. If I have to program it in machine code, could you please give me some instructions on how to use me machine code monitor.
John Evans, Cheshire.

Dear John,
The method for connecting two 64s is explained in detail in Steve Carrie's 65XX Interfacing series. In the April 1990 issue, on page 39, you will find a BASIC program to do what you want, and opposite that there is a diagram illustrating the cable connections. I wish you luck with linking the computers.

TIP OF THE MONTH

This month I am proud to present not one but two tips from various readers. The first comes to you courtesy of David Lomax from Wallasey, Merseyside, and involves the game Limbo 2. Take it away David...

Have you ever wanted to see all of the level screens of a game with out playing through each level? Well as you know it's pretty impossible but I've made it possible for CDU readers playing Limbo 2. However, you will need a reset button. Load Limbo 2, press your reset button and type SYS 1100 and press RETURN. You should see a screen full of garbage. Now press fire. You should now see a clear, still level screen. Press your reset button again and reboot Limbo 2. Repeat this as before except type SYS 1200, then 1300, then 1400 and so on. Each time you will see a different level screen. Continue this until you reach SYS 2000. This is the last screen available.

Thanks for that David. The second one is aimed at all the users of the old type C64s and concerns itself with the CDU Demo that appeared on the March 1990 disk. Stuart Kelly of Argyll, Scotland, brought it to our attention.

Load up the CDU Demo (March 1990) and type in the POKEs shown below. Then run it and you should see some things that you haven't seen before - POKE 18056,33: POKE 18057,208: POKE 20542,33: POKE 20543,208

Thanks Stuart for those POKEs. In case anyone is wondering what the elusive details are, I shall tell you.... Second thoughts - you try it out yourself. Suffice to say it is quite major but it only pertains to the second and third parts. Next month I shall present you with a little machine code routine that we have received that will provide very useful in debugging your BASIC programs.

That marks the end of this month's Techno Info section but I trust that I will see you all again next month. If you are experiencing any problems with your system or different programs then please write to us at CDU Techno Info, 11 Cook Close, Brownsover, Rugby, Warwickshire, CV21 1NG. That is also the address to which you should send your tips if you would like them published in the Tip of the Month section.

FEATURE

65XX INTERFACING

STEVE CARRIE winds up this interesting series with a look at the PLUS/4

BEFORE I LET STEVE CARRY ON. I WOULD JUST LIKE TO PERSONALLY THANK HIM FOR SUCH AN INTERESTING AND EDUCATIONAL SERIES. WELL DONE STEVE.....ED!!
I ALSO MUST APOLOGISE FOR THE LONG LISTINGS. THERE WAS NOT SUFFICIENT ROOM ON THE DISK TO INCLUDE THEM AS SOURCE FILES.

As someone who was brought up on VIC 20's and COMMODORE 64's, I didn't have much contact with the C16 and PLUS/4 until fairly recently. I was disappointed with the lack of backup that these machines received, in particular the PLUS/4 which is an excellent machine.

Even now, these is still very little information available for it although, ANCO publishing have gone a long way to rectifying this. When I was preparing this section, I tried various sources including Commodore themselves to get information on the 6529 single port interface (two of which are inside the PLUS/4), I didn't meet with much success so I spent some considerable time looking at the PLUS/4's printed circuit board, tracing out the tracks and building a picture of what makes this baby tick.

I must say at this point that, of all the companies I called, the most helpful was MOTOROLA SEMICONDUCTORS LTD and my thanks go out to them.

SOME INSIDE INFO

The PLUS/4 is a complex little beast with considerable possibilities. On-board Rom-bank switching software and hardware allows a number of Roms to be banked. The 3+1 software supplied with the machine is an example of this and may be removed by extracting the 3rd and 4th Roms (counting from the left) of the group of chips at the front of the circuit board. You could program your own Eprom's and slot them in instead.

Inside the PLUS/4 you will find, in addition to the 7501 CPU and 8530 TED, two 6529 SPI's and 8551 ACIA (similar to a 6551 in most respects).

What I uncover in this section may quite possibly be news to most of you but it is possible to interface a number of things to the PLUS/4, including a parallel printer (actually used in the preparation of this article) and another machine with full 8-bit parallel transfer. The technique I have used combines facilities from both the 8551 and 6529.

The 6529 SPI is a strange and simple little device. From the programming point of view, it appears to consist of a data input/output register appearing at $FD10 and a data enable register at $FD17. In order for data to be sent or received, the corresponding bit in the enable register must be set to a 1. We would normally write $FF (all bits enabled) to the enable register. Writing to the data register sends data out over the 8 user-port lines and reading the register returns the current state of these lines. Unlike the 6526 the SPI has no dedicated handshaking lines. This makes our job of interfacing to a printer or computer a bit tricky since there is no way of regulating the data flow.

Whilst pondering this little problem, lets look at the ACIA. as you may have seen in the PLUS/4's manual, there are a number of RS232-type lines appearing at the user port. Programming-wise, the ACIA appears as 4 registers at base $FD00. FIG 10 gives the details.

FIG 10.

bit ->	7	6	5	4	3	2	1	0
	IRQ	DSR	DCD	TDRE	RDRE	OVRN	FRE	PAE

FIG 10A: REGISTERS

Base+0	**Data in/out register**
Base+1	**Chip reset/status register**
Base+2	**Command register**
Base+3	**Control register**

FIG 10B: REGISTER DETAILS

Base+01 (FD01) STATUS REGISTER.
Writing to this register resets the acia. Reading this register returns the following format:

IRQ	when 1, Int has occurred
DSR	when 1, DSR line is active
DCD	when 1, DCD line is active
TDRE	when 1, Transmit register is empty
RDRE	when 1, Receiver register is empty
OVRN	when 1, Receiver overrun has occurred
FRE	when 1, Framing error was detected
PAE	when 1, Parity error was detected.

Note: bits 3 thru 6 will cause an interrupt if enabled.

Base+02 (FD02) COMMAND REGISTER

bit ->	7	6	5	4	3	2	1	0
	PC1	PC0	PMD	RCH	TC1	TC0	IRQ	DTR

bits 7-6	parity control (PC0 & PC1)
7 6	**Effect**
0 0	Odd parity tx/rx
0 1	Even parity tx/rx
1 0	Mark parity tx, rx disabled
1 1	Dpace parity tx, rx disabled
	bit5 Parity mode (pmd)
0	Disabled
1	Enabled
	bit4 Receiver echo (RCH)
0	Normal
1	Echo (bits 2/3 must both be zero)
bits 3-2	Transmitter control (TC0 & TC1)
3 2	**Effect**
0 0	RTS high, tx IRQ off
0 1	RTS low, tx IRQ on
1 0	RTS low, tx IRQ off
1 1	RTS low, tx IRQ off (send break)
bit 1	Receiver IRQ control (IRQ)
0	RX IRQ ON
1	RX IRQ OFF
bit 0	Dtr control (DTR)
0	DTR HIGH
1	DTR LOW

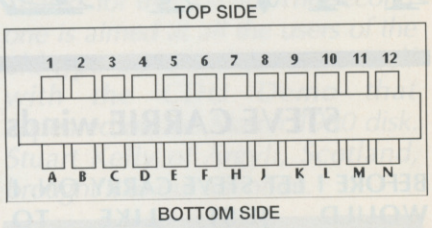
Base+03 (FD03) CONTROL REGISTER

bit ->	7	6	5	4	3	2	1	0
	SBT	WL1	WL0	RCS	BR3	BR2	BR1	BR0

bit 7	Stop bits (SBT)
0	1 stop bit
1	2 stop bits
1	(1.5 stop bits when wl=5, no parity)
1	(1 stop bit when wl=8 and parity)
bits 6-5	word length (wl0 & wl1)
6 5	**Effect**
0 0	8 bits
0 1	7 bits
1 0	6 bits
1 1	5 bits
bit 4	Receiver clock source (RCS)
0	External
1	Internal (see baud rate below)
bits 3-0	baud rate (internal clock) (br3-br0)
3 2 1 0	**baud rate**
0 0 0 0	16x external clock
0 0 0 1	50
0 0 1 0	75
0 0 1 1	109.92
0 1 0 0	134.58
0 1 0 1	150
0 1 1 0	300
0 1 1 1	600
1 0 0 0	1200
1 0 0 1	1800
1 0 1 0	2400
1 0 1 1	3600
1 1 0 0	4800
1 1 0 1	7200
1 1 1 0	9600
1 1 1 1	19200

The PLUS/4's kernal ROM handles the RS232 already and uses a combination of ACIA and SPI registers. If you are planning to use RS232, it's probably best to use these built-in facilities. As you can see, the ACIA has various control lines such as request to send (RTS) and data set ready (DSR) which are part of the RS232 standard. These also appear at the user port and some are programmable so we can use some of these as handshake lines for the parallel port.

Fig 11 shows the pin assignments at the user port. What knowledge I have of this has been gained by looking at the machine's printer circuit board and some guesswork inspired by past electronics training (ex-electronics engineer). What I haven't found out you may be able to fill in.



PIN ASSIGNMENT

TOPSIDE

1	**ground**
4	**6529 data 2**
5	**6529 data 3**
6	**6529 data 4**
7	**6529 data 5**
8	**acia external receive clock**
9	**?????**
10	**9v ac phase +**
11	**9v dc phase -**
12	**ground**

BOTTOMSIDE

a	**ground**
b	**6529 data 0**
c	**acia rxd**
d	**acia RTS**
e	**acia dtr**
f	**6529 data 7**
h	**????**
j	**6529 data 6**
k	**6529 data 1**
l	**acia DSR**
m	**acia txd**
n	**ground**

As you can see, the data port lines are a bit scattered. If you are making a connection to another PLUS/4, then use the wiring in fig 12a. If you connect to a 64 or 128, use fig 12b.

In both cases, I am using the ACIA's request to send (RTS) control line as an data ready/accepted line (as in the CIA's PC output) and the data set ready (DSR) input as a data accepted/ready line (similar to the

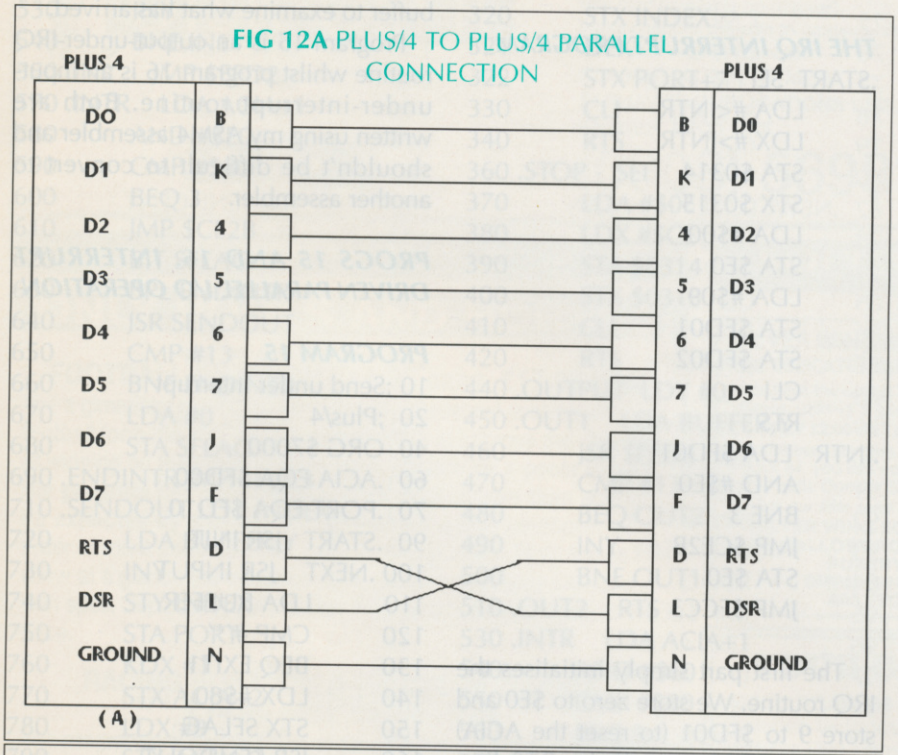


FIG 12A PLUS/4 TO PLUS/4 PARALLEL CONNECTION

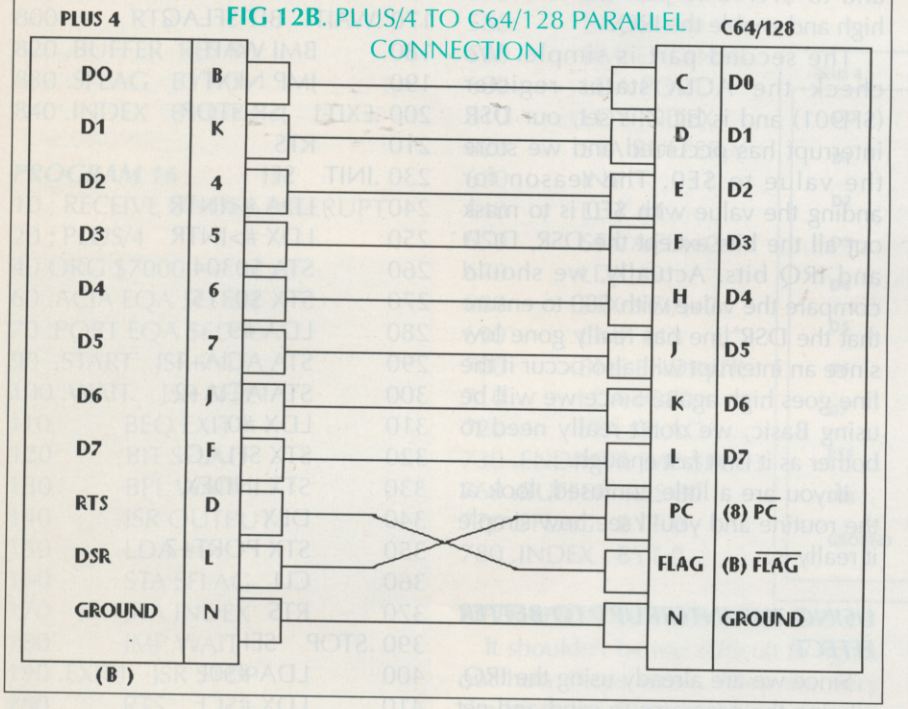


FIG 12B. PLUS/4 TO C64/128 PARALLEL CONNECTION

CIA's FLAG input).

The program needed to run the port must use a short interrupt service routine in order to detect the DSR input which is reliably detected only as an IRQ interrupt. When using the user port like this, the system's RS232 activity should be disabled.

When installed, the interrupt routine will check if it was called by the ACIA and will set a zeropage memory location to $80. A foreground program should poll this location until it reads $80 (128) whereupon it should set it back to zero and continue. Thus we have an event flag which operates like FLAG on the CIA in most respects except that we have to clear it ourselves.

We use the RTS line by pulsing it low. We write to bits 2 and 3 of the command register. This tells a connected device that valid data is present (output) or that data has been accepted (input). Here then, we emulate a strobe output similar to the PC line of the CIA. Using the port from basic.

USING THE PORT FROM BASIC

As with the section on the CIA, I'll begin with a couple of Basic programs to show the basic operation of the port. Programs 13 and 14 begin by pokeing the short interrupt routine into memory, in the RS232 buffer to be precise and initialising it. Memory location $E0 (224) is used as the data flag.

PROGS 13/ 14. SIMPLE PLUS/4 I/O

```
10    GOSUB130
20    DO
30    INPUT R$
40    R$=R$+CHR$(13)
50    FORI=1TOLEN(R$)
60    POKEC<ASC(MID$(R$,I,1))
70    POKEA+2,1:POKEA+2,9
80    DO:LOOPWHILE
      PEEK(224)=0
90    POKE224,0
100   NEXTI
110   LOOPG4
130   RESTORE
140   FORI=0TO47
150   READG$:POKE1015+
      I,DEC(G$)
160   NEXTI
170   C=DEC("FD10")
180   A=DEC("FD00")
190   SYS1015
200   RETURN
220   DATA 78,A9,17,A2
      04,BD,14,03
230   DATA 8E,15,03,A9,09,8D
      01 ,FD
240   DATA 8D,02,FD,A9,00,85,
      E0,58
250   DATA 60,EA,EA,EA,EA,EA,
      EA,EA
260   DATA AD,01,FD,29,E0,D0,
      03,4C
270   DATA 0E,CE,85,E0,4C,C3,
      FC,00
```

```
10    GOSUB90
20    DO
```

```
30      DO:LOOP WHILE
        PEEK(224)=0
40      POKE224,0
50      PRINTCHR$(PEEK(C));
60      POKEA+2,1:POKEA+2,9
70      LOOP
90      RESTORE
100     FORI=0TO47
110     READG$:POKE1015+
        I,DEC(G$)
120     NEXTI
130     C=DEC("FD10")
140     A=DEC("FD00")
150     SYS1015
160     RETURN
180     DATA 78,A9,17,A2,04,8D,
        14,03
190     DATA 8E,15,03,A9,09,8D
        ,01,FD
200      DATA 8D,02,FD,A9,00
        85,E0,58
210     DATA 60,EA,EA,EA,EA,
        EA,EA,EA
220     DATA AD,01,FD,29,
        E0,D0,03,4C
230     DATA 0E,CE,85,E0,4C,
        C3,FC,00
```

PROGRAM 13 OPERATION

After initialisation, the user inputs a string which has a return character (13) added to it. The string is sent out one character at a time between lines 50 and 100. The character is POKEd to the SPI I/O register at $FD10. next, valid data is signalled by pulsing the RTS line low in line 70. This is achieved by clearing bit 3 of the command register and then setting it again. Next, we wait for location 224 to indicate (via the IRQ routine) that the external device has accepted the data.

PROGRAM 14 OPERATION

Once initialised, we begin by waiting for location 224 to indicate that an external device has sent data. When this happens, we clear the location to zero and recover the byte from the data register, then pulse the RTS line low.

THE INTERRUPT ROUTINE

Both of these programs use a simple interrupt routine to detect the DSR line. See fig 13 for an assembly language listing of the program.

THE IRQ INTERRUPT PROGRAM

```
.START  SEI
        LDA #<INTR
        LDX #>INTR
        STA $0314
        STX $0315
        LDA #$00
        STA $E0
        LDA #$09
        STA $FD01
        STA $FD02
        CLI
        RTS
.INTR   LDA $FD01
        AND #$E0
        BNE 3
        JMP $CE2B
        STA $E0
        JMP $FCC3
```

The first part simply initialises the IRQ routine. We store zero to $E0 and store 9 to $FD01 (to reset the ACIA) and to $FD02 to pull the RTS line high and enable the IRQ.

The second part is simple. We check the ACIA status register ($FD01) and is bit 7 is set, our DSR interrupt has occurred and we store the value to $E0. The reason for anding the value with $E0 is to mask out all the bits except the DSR, DCD and IRQ bits. Actually, we should compare the value with $80 to ensure that the DSR line has really gone low since an interrupt will also occur if the line goes high again. Since we will be using Basic, we don't really need to bother as it isn't fast enough.

If you are a little confused, look at the routine and you'll see how simple it really is.

USING THE INTERRUPT TO BETTER EFFECT

Since we are already using the IRQ, altering the programs to send and get data entirely under interrupt isn't all that much different from what we are using already. To send data, we store the information in a buffer and signal to the IRQ that it may begin sending it out. To receive, the interrupt routine recovers data and stores it in the buffer. When all the data has been received, the IRQ signals the main program that it may now access the buffer to examine what has arrived.

Program 15 is an output-under-IRQ routine whilst program 16 is an input-under-interrupt routine. Both are written using my ASM4 assembler and shouldn't be difficult to convert to another assembler.

PROGS 15 AND 16. INTERRUPT DRIVEN PARALLEL I/O OPERATION

PROGRAM 15

```
10 ;Send under Interrupt
20  ;Plus/4
40  ORG $7000
60  .ACIA EQA $FD00
70  .PORT EQA $FD10
90  .START   JSR INIT
100 .NEXT    JSR INPUT
110          LDA BUFFER
120          CMP #'*'
130          BEQ EXIT1
140          LDX #$80
150          STX SFLAG
160          JSR SENDOUT
170 .WAIT    BIT SFLAG
180          BMI WAIT
190          JMP NEXT
200 .EXIT1   JSR STOP
210          RTS
230 .INIT    SEI
240          LDA #<INTR
250          LDX #>INTR
260          STA $0304
270          STX $0315
280          LDA #9
290          STA ACIA+1
300          STA ACIA+2
310          LDX #0
320          STX SFLAG
330          STX INDEX
340          DEX
350          STX PORT+7
360          CLI
370          RTS
390 .STOP    SEI
400          LDA #$0E
410          LDX #$CE
420          STA $0314
430          STX $0315
440          CLI
450          RTS
470 .INPUT   LDY #0
480          STY INDEX
490 .IN1     JSR $FFCF
500          STA BUFFER,Y
510          CMP #13
520          BEQ IN2
```

```
530         INY
540         BNE IN1
550 .IN2    JMP $FFD2
570 .INTR   LDA ACIA+1
580         AND #$E0
590         CMP #$80
600         BEQ 3
610         JMP $CE2B
620         BIT SFLAG
630         BPL ENDINTR
640         JSR SENDOUT
650         CMP #13
660         BNE ENDINTR
670         LDA #0
680         STA SFLAG
690 .ENDINTR JMP $FCC3
710 .SENDOUT LDY INDEX
720         LDA BUFFER,Y
730         INY
740         STY INDEX
750         STA PORT
760         KDX #1
770         STX ACIA+2
780         LDX #9
790         STX ACIA+2
800         RTS
820 .BUFFER  RES 80
830 .SFLAG   BYT 0
840 .INDEX   BYT 0
```

PROGRAM 16

```
10 ; RECEIVE UNDER INTERUPT
20 ; PLUS/4
40 ORG $7000
60 .ACIA EQA $FD00
70 .PORT EQA $FD10
90  .START   JSR INIT
100 .WAIT    JSR $FFE1
110         BEQ EXIT1
120         BIT SFLAG
130         BPL WAIT
140         JSR OUTPUT
150         LDA #0
160         STA SFLAG
170         STA INDEX
180         JMP WAIT
190 .EXIT1   JSR STOP
200         RTS
220 .INIT    SEI
230         LDA #<INTR
240         LDX #>INTR
250         STA $0314
260         STX $0315
270         LDA #9
280         STA ACIA+1
290         STA ACIA+2
300         LDX #0
310         STX SFLAG
320         STX INDEX
321         DEX
322         STX PORT+7
330         CLI
340         RTS
360 .STOP    SEI
370         LDA #$0E
380         LDX #$CE
390         STA $0314
400         STX $0315
410         CLI
420         RTS
440 .OUTPUT  LDY #0
450 .OUT1    LDA BUFFER,Y
460         JSR $FFD2
470         CMP #13
480         BEQ OUT2
490         INY
500         BNE OUT1
510 .OUT2    RTS
530 .INTR    LDA ACIA+1
540         AND #$E0
550         CMP #$80
560         BEQ 3
570         JMP $CE2B
580         BIT SFLAG
590         BMI ENDINTR
600         LDA PORT
610         LDY INDEX
620         STA BUFFER,Y
630         INY
640         LDX #1
650         STX ACIA+2
670         LDX #9
680         STX ACIA+2
690         CMP #13
700         BNE ENDINTR
710         LDA #$80
720         STA SFLAG
730 .ENDINTR JMP $FCC3
760 .BUFFER  RES 80
770 .SFLAG   BYT 0
780 .INDEX   BYT 0
```

OPERATION OF PROGS 15 AND 16

It shouldn't be too difficult to figure out how these work. They are very similar to the CIA serial I/O programs given earlier. In program 15, notice how we must start the data out process ourselves in the main program. We must do this in order to start the interrupt system up. Start either program using SYS dec("7000").

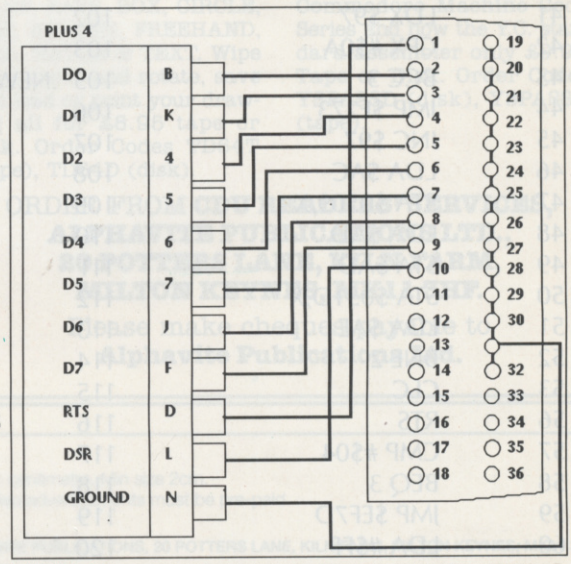
PARALLEL PRINTER TIME!

Program 17 is the PLUS/4 version of the C64 printer driver given earlier. The connecting cable should be wired as shown in fig 14. the major difference between this and the 64 version is the fact that I am using device number 4 instead of 5. I did this in order that the 3+1 software would have a better chance of working with it. I've put the program at $7000 which is perhaps not the best place for it normally, but it should serve OK for the demonstration. Start the program with SYS dec("7000"). Relocate as required.

To use a parallel printer;
OPEN <FILE NO.>,4,<SEC ADDR.>

If the secondary address is 10 then a line feed is sent out after a return. for example;

OPEN 1,4,10 opens the interface with a linefeed after a CR whilst open 1,4 opens it with no line feed after CR.



36 WAY AMPHENOL PLUG WIRING SIDE (PINS 14-32 ALL GND).

PROGRAM 17. PLUS /4 PRINTER DRIVER

```
1   ;Centronics Printer Driver
2   ; Plus/4 version
4   ORG $7000
6   .PORT EQA $FD10
7   .ACIA EQA $FD00
10  .START    SEI
11            LDX #<NEWOPEN
12            LDY #>NEWOPEN
```

```
13          STX $0318
14          STY $0319
15          LDX #<NEWCLOSE
16          LDY #>NEWCLOSE
17          STX $031A
18          STY $031B
19          LDX #<NEWCHKOUT
20          LDY #>NEWCHKOUT
21          STX $031E
22          STY $031F
23          LDX #<NEWCLRCHN
24          LDY #>NEWCLRCHN
25          STX $0320
26          STY $0321
27          LDX #<NEWCHROUT
28          LDY #>NEWCHROUT
29          STX $0324
30          STY $0325
31          CLI
32          RTS
34  .FLAG BYT 0
35  .RTLF BYT 0
37  .NEWOPEN    LDX $AC
38          JSR $EEEB
39          BNE 3
40          JMP $F276
41          LDX $97
42          CPX #$0A
43          BCC 3
44          JMP $F273
45          INC $97
46          LDA $AC
47          STA $0509,X
48          LDA $AD
49          STA $AD
50          STA $051D,X
51          LDA $AE
52          BNE 2
53          CLC
56          RTS
57          CMP #$04
58          BEQ 3
59          JMP $EF7D
60          LDA #$FF
61          STA PORT+7
62          SEI
63          LDA #<INTR
64          LDX #>INTR
65          STA $0314
66          STX $0315
67          LDA #10
68          STA ACIA+1
69          STA ACIA+2
70          LDA #0
71          STA FLAG
72          LDA $AD
73          AND #15
74          LDX #0
75          CMP #10
76          BNE 2
77          LDX #$80
78          STX RTLF
79          CLC
80          RTS
82  .NEWCLOSE   ROR $BA
83          JSR $EEED
84          BEQ 2
85          CLC
86          RTS
87          JSR $EEF8
88          TXA
89          PHA
90          LDA $AE
91          BNE 3
92          JMP $EECA
93          CMP #$04
94          BEQ 3
95          JMP $EE6F
96          LDA #$0E
97          LDX #$CE
98          SEI
99          STA $0314
100         STX $0315
101         STA $FD01
102         CLI
103         JMP $EECA
105 .NEWCHKOUT JSR $EEE8
106         BEQ 3
107         JMP $F279
108         JSR $EEF8
109         BNE 3
110         JMP $F285
111         CMP #$04
112         BEQ 3
113         JMP $ED70
114         STA $99
115         PHA
116         LDA #$FF
117         STA PORT+7
118         PLA
119         CLC
120         RTS
122 .NEWCLRCHN LDX #$04
123         CPX $99
124         BEQ 3
125         JMP $EF0C
126         LDX #$03
127         STX $99
128         RTS
130 .NEWCHROUT PHA
131         LDA $99
132         CMP #$04
133         BEQ 3
134         JMP $EC4C
135         PLA
136         JSR SENDCENT
137         CMP #13
138         BNE NOLF
139         BIT RTLF
140         BPL NOLF
141         PHA
142         LDA #$10
143         JSR SENDCENT
144         PLA
145 .NOLF     CLC
146         RTS
148 .SENDCENT  PHA
149         STA PORT
150         LDA #2
151         STA ACIA+2
152         LDA #9
153         STA ACIA+2
154 .WAIT     BIT FLAG
155         BPL WAIT
156         LDA #0
157         STA FLAG
158         PLA
159         RTS
161 .INTR     LDA ACIA+1
162         AND #$E0
163         CMP #$80
164         BEQ 3
165         JMP $CE0E
166         STA FLAG
167         LDA #10
168         STA ACIA+2
169         JMP $FCC3
```

That concludes our brief look at the PLUS/4's parallel interface. It has to be said that most of us (including myself) thought that the machine had no such facility, but then I know I've been wrong on numerous occasions when it comes to the PLUS/4. I'd be happy to hear from any of you who go on to use the interface for other applications. One that came to mind was a BASIC program transfer system which would be useful to C64 and PLUS/4 owners who use tape drives which can't read each others tapes (incompatibility rules supreme as usual...). Well I hope you have found this series useful and are a bit more familiar with these underused, but nevertheless, useful devices.

SEND ANY COMMENTS ABOUT THIS SERIES TO THE EDITORIAL OFFICE, MARK IT F.A.O. STEVE CARRIE. THANK YOU.

Commodore DISK USER

Lineage: 53p per word. (+VAT)
Semi display: £11.50 plus VAT per single column centimetre minimum 2cm.
Ring for information on series bookings/discounts.
All advertisements in this section must be prepaid.
Advertisements are accepted subject to the terms and conditions printed on the advertisement rate card (available on request).
Make cheques payable to Alphavite Publications Ltd.
Send your requirements to:
CLASSIFIED DEPARTMENT
ALPHAVITE PUBLICATIONS LTD., 20 POTTERS LANE, KILN FARM, MILTON KEYNES, MK11 3HF.

ACCESS BARCLAYCARD VISA

(0908) 569819

SOFTWARE

ONLY POOLS AND HORSES

FOOTBALL BOXFORM £14.95 cass £17.95 disk

Not a gimmicky plaything but a genuine statistical analysis of football form the sole objective of which is to improve the chances of winning on the pools or fixed odds. **Written by a pools expert the program has forecast over 50% more draws than would be expected by chance.** Homes, aways & draws are shown in order of merit and true odds given for every match. Merit tables show at a glance the teams currently in form and those having a lean spell. Separate AUSTRALIAN POOLS program included in the price.

RACING BOXFORM & HANDICAP WINNER £14.95 cass £17.95 disk

Two programmes for the price of one. Boxform weighs up a race by considering the many facets of form. Easily amended to suit the user's ideas. HANDICAP WINNER shows the value for money bets. Over 1000 winners every year and ON DERBY DAY 1990 THE PROGRAM GAVE THE WINNERS OF 7 OUT OF 9 HANDICAPS. 20/1 10/1 7/1 11/2 4/1 4/1 6/4. **PROBABLY THE BEST DAY'S RESULTS EVER ACHIEVED, WITH OR WITHOUT A COMPUTER.**

SPECIAL OFFER: All the above for £24.95 (disks £29.95) plus a FREE program to work out almost any bet. So good it's used by bookies.

Why pay inflated prices? **BOX**oft **CLEVER** the same – get the **BEST**.
Advertised for five years in the sporting and computer press.
BOXOFT (CDU), 65 Allans Meadow, Neston,
South Wirral, L64 9SQ
Cheque/P.O./Access/Visa Tel: 051-336-2668 (24 hrs)

SOFTWARE

SILVER WING SOFTWARE
'THE DEDICATED C64 P.D. LIBRARY

We have the best & latest quality demos, utilities & games available.
ONLY £2.00 on Tape or Disk
For the best benefits & power from your 64, you can not do without Public Domain Software...
For our lists send an SAE to:
SILVER WING SOFTWARE
185 CALLOWBROOK LANE,
RUBERY,
BIRMINGHAM
B45 9TG.

SERVICES

COMMODORE SUPPLIES

C64 Power Unit	£21.99
C128 Power Unit	£39.99
C2N/C3N Datasette	£24.99
100 x 3½" Discs	£29.99
C64 Magnum Lightgun	£34.99
C64/128 Serial Cable	£5.99

Prices include VAT & P&P Chq/PO to:
OMNIDALE SUPPLIES (CD),
23 Curzon Street, Derby, DE1 2E6
Tel 0332 291219

YOUR COMMODORE SPECIALS

TECH DRAW 64 – A comprehensive technical illustration aid for C64 owners for both disk and tape. Commands available include LINE DRAW, RAYS, BOX, CIRCLE, ARC, ELLIPSE, FREEHAND, FILL, HATCH & TEXT. Wipe copy, move and rotate, save and load or print your drawing all for £8.95 tape or disk. Order Codes TD64T (tape), TD64D (disk).

SPEEDY ASSEMBLER – Your very own Assembler, a 100% memory resident program for loading from tape or disk, featured in the Your Commodore Machine Code Series and now the Y.C. standard assembler only £9.95 Tape or Disk. Order Codes YSPASSD (disk), YSPASSR (tape).

ORDER FROM **CDU READERS' SERVICES,**
ALPHAVITE PUBLICATIONS LTD.,
20 POTTERS LANE, KILN FARM,
MILTON KEYNES, MK11 3HF.
Please make cheques payable to
Alphavite Publications Ltd.

Commodore DISK USER

Lineage: **53p** per word. (+VAT)
Semi display: **£11.50** (+ VAT) per single column centimetre. Min size 2cm.
Series discounts available on request. All classified advertisements must be pre-paid.
There are no reimbursements for cancellations.
SEND TO CDU CLASSIFIED DEPARTMENT, ALPHAVITE PUBLICATIONS, 20 POTTERS LANE, KILN FARM, MILTON KEYNES, MK11 3HF.

I enclose my cheque/postal order for £ ..... for ..... insertions, made payable to Alphavite Publications.
(Delete as necessary)
PLEASE DEBIT MY ACCESS/VISA CARD NO:
EXP. DATE. .....
£ ..... FOR ..... INSERTIONS
Name .....
Address ..... Postcode .....
Daytime Tel No ..... Signature ..... Date .....

O FOR SALE **O** SOFTWARE **O** SPECIAL OFFERS **O** REPAIRS **O** HARDWARE **O** DISKS

...it's dynamite!

POWER CARTRIDGE

FOR YOUR COMMODORE 64/128

"...unbelievable value for money" ZZAPP! Dec 89

- POWER TOOLKIT
- POWER MONITOR
- TAPE & DISK TURBO
- PRINTERTOOL
- POWER RESET
- TOTAL BACKUP

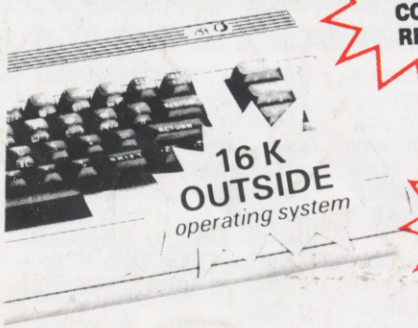
..."Money well spent" YC/CDU Jan 90

42 Pg Manual - "Damned Good Handbook" CCI Jan 90

SO MUCH FOR SO LITTLE

AVAILABLE FROM ALL GOOD COMPUTER RETAILERS

TRIED AND TESTED - OVER 100,000 SOLD IN EUROPE



..."highly recommended for C64 users" CCI Jan 90

YOU WILL WONDER HOW YOU EVER MANAGED WITHOUT IT

POWER TOOLKIT

A powerful BASIC-Toolkit (Additional helpful commands) that considerably simplifies programming and debugging.

AUTO	HARDCAT	RENUMBER
AUDIO	HARDCOPY	REPEAT
COLOR	HEX$	SAFE
DEEK	INFO	TRACE
DELETE	KEY	UNNEW
DOKE	PAUSE	QUIT
DUMP	PLIST	MONITOR
FIND	ILOAD	BLOAD

RENUMBER : *Also modifies all the GOTO's GOSUB's etc. Allows part of a program to be renumbered or displaced.*
PSET : *Set up of printer type.*
HARDCAT : *Prints out Directory.*

The toolkit commands can be used in your programs.

DISK TOOL

Using POWER CARTRIDGE you can load up to 6 times faster from disk.
The Disk commands can be used in your own programs.

DLOAD	DVERIFY	DIR
DSAVE	MERGE	DEVICE
DISK		

MERGE : *Two BASIC programs can be merged into one.*
DISK : *With DISK you can send commands directly to your disk.*

TAPE TOOL

Using POWER CARTRIDGE you can work up to 10 times faster with your data recorder. The Tape commands can be used in your own programs.

LOAD	SAVE	VERIFY
MERGE	AUDIO	

POWERMON

A powerful machine language monitor that is readily available and leaves all of your Commodore memory available for programming.
Also works in BASIC-ROM, KERNAL and I/O areas.

A ASSEMBLE	I INTERPRET	S SAVE
C COMPARE	J JUMP	T TRANSFER
D DIS-ASSEMBLE	L LOAD	V VERIFY
	M MEMORY	W WALK
F FILL	P PRINT	X EXIT
G GO	R REGISTER	$ DIRECTORY DOS Commands
H HUNT		

PRINTERTOOL

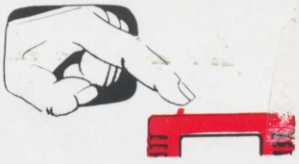
The POWER CARTRIDGE contains a very effective Printer-Interface, that self detects if a printer is connected to the Serial Bus or User-Port.
It will print all Commodore characters on Epson and compatible printers.
The printer-interface has a variety of set-up possibilities. It can produce HARDCOPY of screens not only on Serial printers (MPS801, 802, 803 etc) but also on Centronic printers (EPSON, STAR, CITIZEN, PANASONIC, etc).
The HARDCOPY function automatically distinguishes between HIRES and LORES. Multi-colour graphics are converted into shades of grey. The PSET functions allow you to decide on Large/Small and Normal/Inverse printing.
The printer PSET functions are:

PSET 0 - *Self detection Serial/Centronics.*
PSET 1 - *EPSON mode only.*
PSET 2 - *SMITH-CORONA mode only.*
PSET 3 - *Turns the printing 90 degrees!!*
PSET 4 - *HARDCOPY setting for MPS802/1526.*

PSET B - *Bit-image mode.*
PSET C - *Setting Lower/Upper case and sending Control Codes.*
PSET T - *All characters are printed in an unmodified state.*
PSET U - *Runs a Serial printer and leaves the User-port available.*
PSET Sx - *Sets the Secondary address for HARDCOPY with Serial Bus.*
PSET L1 - *Adds a line-feed, CHR$ (10), after every line.*
PSET L0 - *Switches PSET L1 off.*

Bitcon Devices Ltd does not authorise or purport to authorise the making by any means or for any purpose whatsoever of copies or adaptions of copyright works or other protected material, and users of the Power Cartridge must obtain the necessary prior consent for the making of such copies or adaptions from all copyright and other right owners concerned. See UK Copyright, Designs & Patents Act 1988.

POWER RESET



On the back of the POWER CARTRIDGE there is a Reset Button. Pressing this button makes a SPECIAL MENU appear on the screen.
This function will work with many programmes.

CONTINUE - *Allows you to return to your program.*
BASIC - *Return to BASIC.*
RESET - *Normal RESET.*
TOTAL BACKUP DISK - *Saves the contents of the memory onto a Disk. The program can be reloaded later with BLOAD followed by CONTINUE.*
RESET ALL TOTAL BACKUP TAPE - *RESET of any program. As BACKUP DISK but to TAPE.*
HARDCOPY - *At any moment, prints out a Hardcopy of the screen. Using CONTINUE afterwards you can return to the program.*
MONITOR - *Takes you into the Machine language Monitor.*

BDL
Bitcon Devices Ltd

88 BEWICK ROAD
GATESHEAD
TYNE AND WEAR
NE8 1RS
ENGLAND

Tel: 091 490 1975 and 490 1919 Fax 091 490 1918
To order: Access/Visa welcome - Cheques or P/O payable to BDL
Price: £16.99 incl. VAT.
UK orders add £1.20 post/pack total - £18.19 incl. VAT.
Europe orders add £2.50. Overseas add £3.50
Scandinavian Mail Order and Trade enquiries to: Bhiab Elektronik, Box 216, Norrtalje 76123, SWEDEN. Tel: + 46 176 18425 Fax: 176 18401
TRADE AND EXPORT ENQUIRIES WELCOME